

Diplomarbeit

zur Erlangung des akademischen Grades
Diplom-Wirtschaftsinformatiker
am Fachgebiet Wirtschaftsinformatik für Industriebetriebe

Automatisierte Generierung von Simulationsmodellen unter Verwendung des Core Manufacturing Simulation Data (CMSD) Information Model - Implementierung eines Pilotszenarios

Betreuer:	Dipl.-Wirtsch.-Inf. Sören Bergmann
Hochschullehrer:	Univ.-Prof. Dr. Steffen Straßburger
Bearbeiter:	Alexander Fiedler 33676

Diese Arbeit wurde am 13.09.2010 in der Fakultät für Wirtschaftswissenschaften der Technischen Universität Ilmenau eingereicht.

URN: urn:nbn:de:gbv:ilm1-2010200290

Eidesstattliche Erklärung

Ich erkläre hiermit, dass ich diese Diplomarbeit selbstständig ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe. Alle den benutzten Quellen wörtlich oder sinngemäß entnommenen Stellen sind als solche einzeln kenntlich gemacht.

Diese Arbeit wurde bislang keiner anderen Prüfungsbehörde vorgelegt und ist auch nicht veröffentlicht worden.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Arnstadt, den 13.09.2010

Alexander Fiedler

Inhaltsverzeichnis

Liste verwendeter Abkürzungen	VI
1 Einleitung	1
1.1 Ausgangssituation und Problemstellung	1
1.2 Zielstellung	2
2 Unterstützung der Planung und Steuerung von Fertigungssystemen durch Simulation	3
2.1 Fertigungsplanung.....	3
2.2 Produktionsplanung und -steuerung	5
2.3 Die Ablaufsimulation im fertigungstechnischen Umfeld	7
2.3.1 Begriffsbestimmung und Definition.....	7
2.3.2 Anwendungsgebiete der Ablaufsimulation	8
2.3.3 Nutzen der Simulation.....	12
2.3.4 Phasen einer Simulationsstudie.....	13
2.3.5 Eingangsdaten für die Simulation	14
2.4 Die Digitale Fabrik	17
2.4.1 Begriffsbestimmung und Definition.....	17
2.4.2 Komponenten der Digitalen Fabrik.....	19
2.4.3 Ablaufsimulation in der Digitalen Fabrik	20
2.4.4 Ziele der Digitalen Fabrik	21
3 Automatisierte Simulationsmodellgenerierung	23
3.1 Voraussetzungen und Ziele	23
3.2 Anwendungsfelder von automatisierten Modellgeneratoren	24
3.2.1 Planungsbegleitender Simulationseinsatz	25
3.2.2 Betriebsbegleitender Simulationseinsatz.....	26
3.3 Stand der Forschung zu automatisierten Modellgeneratoren.....	27
3.4 Fazit.....	31
4 Integration der Simulation im betrieblichen Umfeld durch das Core Manufacturing Simulation Data (CMSD) Information Model.....	32
4.1 Motivation und Zielsetzung	32
4.2 Aufbau und Struktur	34
4.2.1 CMSD Package	35
4.2.2 Support Package.....	35
4.2.3 Layout Package	38
4.2.4 Production Operations Package	40
4.2.5 Production Planning Package.....	40

4.2.6	Resource Information Package.....	41
4.3	Beitrag zur Integration automatisierter Modellgeneratoren.....	42
4.3.1	Abbildung statischer Zusammenhänge	43
4.3.2	Abbildung dynamischer Zusammenhänge	44
4.4	Fazit.....	45
5	Prototypische Umsetzung eines automatisierten Modellgenerators.....	46
5.1	Integration in das Umfeld betrieblicher Anwendungssysteme	46
5.2	Struktur des Modellgenerators	48
5.2.1	Bausteinbibliothek.....	49
5.2.2	Datenimport.....	50
5.2.3	Datenhaltung	51
5.2.4	Modellerstellung.....	51
5.3	Anwendungsszenario	52
5.4	Struktur der CMSD-Eingangsdaten	53
5.4.1	Layout Package	54
5.4.2	Production Operations Package	55
5.4.3	Production Planning Package.....	56
5.4.4	Resource Information Package.....	57
5.5	Fazit.....	58
6	Vorschläge für Erweiterungen	60
6.1	Abbildung von Strategien durch „Microfunktionen“	60
6.2	Online-Kopplung zur Produktionssteuerung	61
7	Zusammenfassung.....	62
	Literaturverzeichnis	64
	Abbildungsverzeichnis.....	69
	Tabellenverzeichnis.....	71
	Anhang.....	72
A	Ausgewählte Klassendefinitionen des CMSD Information Model - UML-Teil	72
A.1	Support Package.....	72
A.2	Production Operations Package	74
A.3	Production Planning Package	75
A.4	Resource Information Package	76
B	Ausgewählte Importmethoden des Prototypen	77
B.1	Methode zum Einlesen des Arbeitsplanes.....	77
B.2	Methode zum Einlesen der Ressourcen und Materialflussobjekte	78
B.3	Methode zum Einlesen des Schichtkalenders	82

C	Ausgewählte Strukturen der CMSD-Eingangsdaten des Beispielszenarios in XML ..	84
C.1	Auszug zur Definition einer Maschine	84
C.2	Auszug zur Definition der Rüstmatrix einer Maschine.....	84

Liste verwendeter Abkürzungen

2D / 3D	Zwei- / Dreidimensional
AR	Augmented Reality
BDE	Betriebsdatenerfassung
CAD	Computer Aided Design
CAPP	Computer Aided Process Planning
CMSD	Core Manufacturing Simulation Data
DES	Discrete Event Simulation
DMU	Digital Mock Up
ERP	Enterprise Resource Planning
FIFO	First In, First Out
FMEA	Failure Mode and Effects Analysis
IT	Informationstechnologie
KOZ	Kürzeste Operationszeit
KRB	Kürzeste Restbearbeitungszeit
MES	Manufacturing Execution System
MPS	Master Production Schedule
NC	Numerical Control
ODBC	Open Database Connectivity
PLM	Product Lifecycle Management
PPS	Produktionsplanung und -steuerung
RC	Robot Control
SISO	Simulation Interoperability Standards Organization
UML	Unified Modeling Language
VDI	Verein Deutscher Ingenieure
VR	Virtual Reality
XML	Extensible Markup Language
XPath	XML Path Language

1 Einleitung

1.1 Ausgangssituation und Problemstellung

Das Umfeld produzierender Unternehmen ist schnellen und oft nur ungenau prognostizierbaren Marktveränderungen ausgesetzt. Als ursächlich hierfür werden veränderte Rahmenbedingungen, wie die gestiegenen Kundenanforderungen, der zunehmende Wettbewerb durch die Globalisierung der Märkte sowie strengere Gesetze und Umweltauflagen angesehen. Mit dem Übergang vom Verkäufermarkt zum Käufermarkt treten Kundenanforderungen hinsichtlich einer größeren Individualität sowie Funktionalität der Produkte bei hoher Qualität und der Wunsch nach umweltschonenden Produkten in den Vordergrund [HeMi03, S.526; Ze06, S.1].

Um die Wettbewerbsfähigkeit der Unternehmen zu erhalten, stieg die Anzahl der angebotenen Produktvarianten in den letzten Jahrzehnten sehr stark an. Ein Industriezweig, der im besonderen Maße diesen Veränderungen gegenübersteht, ist die Automobilindustrie. Hier führte die Individualisierung der Nachfrage seit Anfang der Neunzigerjahre zu einem enormen Anstieg der Fahrzeugvarianten bei gleichzeitiger Verkürzung der Produktlebenszyklen [Ro09, S.1].

Eine schnelle Anpassung an die veränderten Anforderungen sowie eine ständige Verbesserung der Fabrik- und Fertigungsplanung kann erfolgsentscheidend sein und stellt die Unternehmen vor große Herausforderungen. Um zeitnah auf mögliche Markttrends reagieren zu können, wird zunehmend versucht, die Entwicklungszeiten von Produkt und Produktionsprozess zu senken. Große Hoffnungen zielen hierbei auf die parallele Entwicklung und Optimierung der Produkte und der Produktionseinrichtungen. Dieses Konzept wird im Rahmen der sogenannten Digitalen Fabrik in der Art verfolgt, dass bereits in der Planungsphase ein detailliertes Abbild der realen Fabrik und ihrer Prozesse entsteht. Hierdurch soll eine beschleunigte und kostengünstigere Planung sowie eine Qualitätssteigerung der Produkte und der Planung erreicht werden.

Im realen Umfeld wirken zahlreiche dynamische und stochastische Einflüsse auf das Produktionssystem ein. Diese dürfen beim digitalen Abbild möglichst nicht unberücksichtigt bleiben. Ein etabliertes Werkzeug sie zu modellieren, ist die Ablaufsimulation¹, welche einen wichtigen Bestandteil der Digitalen Fabrik darstellt. Die heutzutage in der Regel noch anzutreffende manuelle Erstellung von Simulationsstudien verhindert jedoch den konsequenten Einsatz, da der Aufwand, trotz aller Verbesserungen der letzten Jahre, immer noch zu groß ist [Fr07, S.1f].

Um diese Lücke zu schließen, entwickelte man den Ansatz Simulationsmodelle automatisiert zu erstellen. Hierzu wurden in der jüngeren Vergangenheit zunehmend Umsetzungen

¹ Der Begriff Materialflusssimulation wird hierzu in der Literatur vielfach synonym verwendet.

publiziert. Ein Problem, welches sich dabei herauskristallisierte, ist die noch unzureichende Schnittstellenanbindung der Simulationsapplikationen an vorhandene Planungssysteme und Fertigungssteuerungssysteme. Von den Autoren werden hier teils sehr unterschiedliche Wege eingeschlagen. Dennoch lässt sich ein Trend zur Verwendung des systemneutralen Austauschformates Extensible Markup Language (XML) erkennen.

Um das Problem der Schnittstellenanbindung anzugehen, entstand in mehrjähriger Arbeit das „Core Manufacturing Simulation Data“ (CMSD) Information Model. Es definiert eine Spezifikation für Kerndaten von Fertigungssimulationen. Für den Datenaustausch wurde ebenfalls das XML-Datenformat ausgewählt.

1.2 Zielstellung

Das Ziel des theoretischen Teils dieser Arbeit ist, zu einer Einschätzung zu kommen, ob das CMSD Information Model als Spezifikation für ein Datenaustauschformat grundsätzlich zur automatisierten Erstellung von Simulationsmodellen eingesetzt werden kann. Hierfür sind die Möglichkeiten zur Abbildung simulationsrelevanter Eingangsdaten zu untersuchen sowie Stärken und Schwächen des Standards zu identifizieren, um Aussagen über die Nutzbarkeit tätigen zu können.

Mit dem praktischen Teil dieser Arbeit wird die Zielstellung verfolgt, die zuvor getroffenen theoretischen Betrachtungen zu vertiefen und eine prototypische Umsetzung eines automatisierten Modellgenerators zu erstellen. Für den Prototypen ist ein Beispielszenario zu formulieren, welches ein Integrationskonzept für das Szenario in die Systemwelt umgebender betrieblicher Anwendungssysteme beinhaltet. Des Weiteren ist zur Demonstration des Prototypen ein konkretes Anwendungsbeispiel einer Fertigung zu entwickeln. Die Datenversorgung des Modellgenerators ist mittels des Datenformates sicherzustellen, welches durch das CMSD Information Model spezifiziert wird.

2 Unterstützung der Planung und Steuerung von Fertigungssystemen durch Simulation

In diesem Kapitel werden zunächst wesentliche Begriffe eingeführt, welche im Rahmen dieser Arbeit von zentraler Bedeutung sind. Die Ablaufsimulation und das Konzept der Digitalen Fabrik sollen hierbei im Mittelpunkt stehen. Zu Beginn beider Unterkapitel wird eine Definition und Begriffseingrenzung vorgenommen. Anschließend folgen im Fall der Simulation Erläuterungen zu den Anwendungsfeldern und Phasen einer Simulationsstudie sowie zu den benötigten Eingangsdaten. Zur Digitalen Fabrik werden, aufbauend auf der Begriffseingrenzung, Ausführungen bezüglich ihrer Komponenten, der Rolle der Ablaufsimulation innerhalb der Digitalen Fabrik und deren Ziele dargelegt.

2.1 Fertigungsplanung

Für die Planung von Produktionseinrichtungen lässt sich in der Literatur keine einheitliche Begriffsbestimmung finden. Neben dem hier verwendeten Begriff der Fertigungsplanung sind weiterhin Arbeitsplanung, Fertigungsvorbereitung und Produktionsplanung häufig anzutreffende Bezeichnungen. Zudem bildeten sich teils völlig unterschiedliche Auffassungen über inhaltliche Aspekte dieser heraus. So ist es nicht unüblich, den Planungsprozess zur Errichtung von Produktionsanlagen als Produktionsplanung zu bezeichnen, während der Begriff für die Planung des laufenden Produktionsprogrammes ebenso Verwendung findet. Deshalb ist es erforderlich, eine konsistente Definition für diese Arbeit zu formulieren. Zunächst soll jedoch geklärt werden, was allgemein unter Planung zu verstehen ist. Nach Wiendahl versteht man unter **Planung**:

„das gedankliche Durchdringen eines zukünftigen Geschehens mit dem Ziel, diejenigen Entscheidungen vorzubereiten und zu treffen, die zur Erreichung des gewünschten Zustandes notwendig sind. Der gewünschte Zustand wird unter Beachtung denkbarer zukünftiger Entwicklungen, die z. T. mit Hilfe von Prognoseverfahren ermittelt werden, als Zielsetzung definiert. Hauptmerkmal der Planung ist demnach die Zukunftsbezogenheit sowie das im Gegensatz zur Improvisation rationale, nachvollziehbare Vorgehen“ [Wi05, S.15].

Unter **Fertigung** soll gemäß Dangelmaier ein grundsätzlich diskontinuierlicher Prozess verstanden werden, der alle technischen Maßnahmen zur Herstellung von Material oder Erzeugnissen beinhaltet [Da01, S.3].

Zur Fertigung im engeren Sinne werden die Teilefertigung, Montage und die Prüfung von Teilen gezählt. Im Laufe der Fertigung erhalten die Einzelteile ihre geometrisch vorgeschriebene Gestalt und weitere vorgegebene Eigenschaften, wie z. B. bestimmte Festigkeitskenn-

werte und Oberflächenbeschaffenheiten. Anschließend werden die gefertigten Teile in der Montage zu komplexen Teilen (Baugruppen, Produkten) zusammengefügt [WaWe06, S.4].

„Ein **Fertigungssystem** besteht aus (elementaren) Arbeitssystemen, die die kleinste Einheit einer Kombination der Potentialfaktoren Betriebsmittel und Arbeitskräfte darstellen und eine oder mehrere Klassen von Transformationen durchführen können“ [Da01, S.5]. Ein Fertigungssystem bzw. ein System interagierender Fertigungssysteme ist das Ergebnis der Fabrik- und Fertigungsplanung [Fr07, S.6]. Im Rahmen dieser Arbeit werden unter **Fertigungsplanung** „alle einmalig zu treffenden Maßnahmen bezüglich der Gestaltung eines Fertigungssystems und der darin stattfindenden Fertigungsprozesse“ (vgl. ebenda S. 5) verstanden.

Aufgaben der Fertigungsplanung (Arbeitsplanung)	
Arbeitsablaufplanung (Prozessgestaltung)	Arbeitssystemplanung (Fertigungsmittelgestaltung)
Planungsvorbereitung <ul style="list-style-type: none"> • Beratung der Konstruktion • Grobplanung 	Materialplanung <ul style="list-style-type: none"> • Lagerartenplanung • Lagerortplanung
Stücklistenverarbeitung <ul style="list-style-type: none"> • Erstellen der Fertigungs- und Montagestücklisten 	Methodenplanung <ul style="list-style-type: none"> • Planung und Entwicklung neuer <ul style="list-style-type: none"> - Planungsmethoden - Planungsunterlagen - Fertigungsmethoden - Fertigungsverfahren • Arbeitsplatzgestaltung • Arbeitsplatzbewertung • Arbeitsablaufstudien
Arbeits-/ Montageplanerstellung <ul style="list-style-type: none"> • Ausgangsteilbestimmung • Arbeitsvorgangsfolgebestimmung • Fertigungsmittelzuordnung • Vorgabezeitbestimmung 	
NC/RC-Programmierung <ul style="list-style-type: none"> • Operationsplanung • Erstellen der Teileprogramme 	
Betriebsmittelplanung <ul style="list-style-type: none"> • Entwicklung von Sonderbetriebsmitteln 	<ul style="list-style-type: none"> • Planung von Fertigungs- und Montagesystemen
Kostenplanung <ul style="list-style-type: none"> • Kalkulation • Verfahrensvergleich • Wirtschaftlichkeitsrechnung 	Technische Investitionsplanung
Produktorientiertes Qualitätsmanagement <ul style="list-style-type: none"> • Prüfplanerstellung 	<ul style="list-style-type: none"> • FMEA-Durchführung

Abbildung 1: Aufgaben der Fertigungsplanung (Arbeitsplanung) nach [Ev96, S.74]

Die Aufgaben der Fertigungsplanung² sind in Anlehnung an Eversheim u. a. die Planung des Arbeitsablaufes, der Arbeitsstätten, der Fertigungsmittel, die Personalplanung und die Kostenkalkulation [Ev02, S.6-13]. Einen Überblick über weitere Aufgaben bietet Abbildung 1.

2.2 Produktionsplanung und -steuerung

Die mit der Produktionsplanung und -steuerung³ (PPS) verfolgten Ziele müssen sich in das Zielsystem des Unternehmens einordnen, da die PPS Bestandteil der Unternehmensplanung ist [Ku05, S.8]. Als Akteur in marktwirtschaftlichen Systemen gilt für Unternehmen in aller Regel das Prinzip der Wirtschaftlichkeit. Dieses erfordert, dass der Quotient aus erbrachter Leistung (Output) und den dafür entstandenen Kosten (Input) möglichst groß sein soll. Demnach sind die Maßnahmen, die im Rahmen der Produktionsplanung und -steuerung in produzierenden Unternehmen zu treffen sind, so zu gestalten, dass eine vorgegebene Leistung mit möglichst niedrigen Kosten erbracht wird (vgl. ebenda S. 9).

Da aber im Allgemeinen keine umfassenden Kosteninformationen zu den Zeitpunkten zur Verfügung stehen an denen die Maßnahmen geplant oder getroffen werden, arbeitet man häufig mit Ersatzzielgrößen, die im Zusammenhang mit den Kostenzielen stehen. Klassische Ersatzzielgrößen sind (vgl. ebenda S. 9f):

- Minimierung der Durchlaufzeiten der Aufträge
- Maximierung der Kapazitätsauslastung der Ressourcen (z. B. Mitarbeiter, Maschinen)
- Minimierung der Terminüberschreitung der Aufträge
- Minimierung der Bestände von Material, Zwischen- und Endprodukten

Zusammenfassend ausgedrückt ist das Ziel der Produktionsplanung und -steuerung, eine nach dem Wirtschaftlichkeitsprinzip gestaltete Produktion bzw. Fertigung zu ermöglichen [Se05, S.7].

Die Aufgabe der **Produktionsplanung und -steuerung** ist es, „das laufende Produktionsprogramm in regelmäßigen Abständen im voraus nach Art und Menge für mehrere Planungsperioden zu planen und unter Beachtung gegebener oder zu planender Kapazitäten, trotz unvermeidlicher Störungen wie Personalausfall, Lieferverzögerungen oder Ausschuß, möglichst gut zu realisieren“ [Wi05, S.253].

Somit setzt die PPS nach der Errichtung der Fertigungsanlagen ein und beschäftigt sich „mit der operativen, zeitlichen und mengenmäßigen Terminierung, Steuerung und Kontrolle und

2 Eversheim verwendet den Begriff Arbeitsplanung und unterteilt diesen in die Arbeitsablaufplanung für den kurzfristigen und in die Arbeitssystemplanung für den langfristigen Zeithorizont [Ev02, S.7].

3 Die Begriffe Fertigungssteuerung und Arbeitssteuerung werden synonym für die Produktionsplanung und -steuerung verwendet [Da09, S.9f; WaWe06, S.4]. In dieser Arbeit wird aufgrund der größeren Verbreitung die Bezeichnung Produktionsplanung und -steuerung (PPS) bevorzugt.

damit zusammenhängend auch mit der Verwaltung aller Vorgänge“ [Ze06, S.15] in einem existierenden Fertigungssystem (vgl. Abbildung 2).

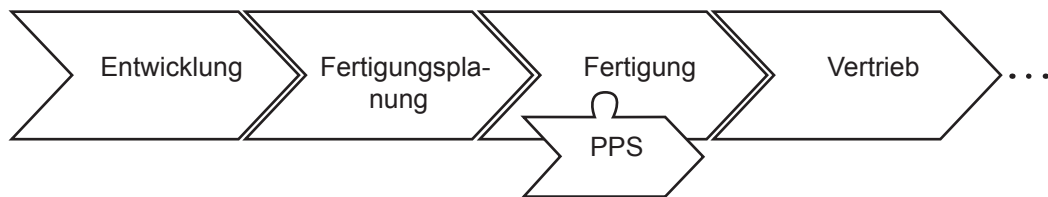


Abbildung 2: PPS als Baustein innerhalb der Fertigung nach [Ze06, S.15]

Der Produktionsplanung werden die tendenziell planerisch orientierten Aufgaben mit längerem Zeithorizont zuteil. Ihre Aufgabe ist es, „die Produktionsaufgaben nach Artikeln, Losgrößen und Wertseinheiten zu bestimmen und auf die produktiven Teilsysteme aufzuteilen“ [Ec02, S.8].

Die Produktionssteuerung übernimmt hingegen operative Aufgaben und präzisiert die von der Produktionsplanung gesetzten Vorgaben. Sie koordiniert das Zusammenwirken der Prozesselemente Arbeitsgegenstand, Arbeitsmittel und Arbeitskraft in ihrer Gesamtheit und unter Berücksichtigung der für die Funktionserfüllung relevanten Merkmale, Zustände und Beziehungen. Zusätzlich gehören die Sicherung der Prozessvoraussetzungen sowie die Koordination von Haupt- und Hilfsprozessen zu ihren Aufgaben [Go89, S.21]. Die wesentlichen Funktionen der PPS sind in Abbildung 3 dargestellt.

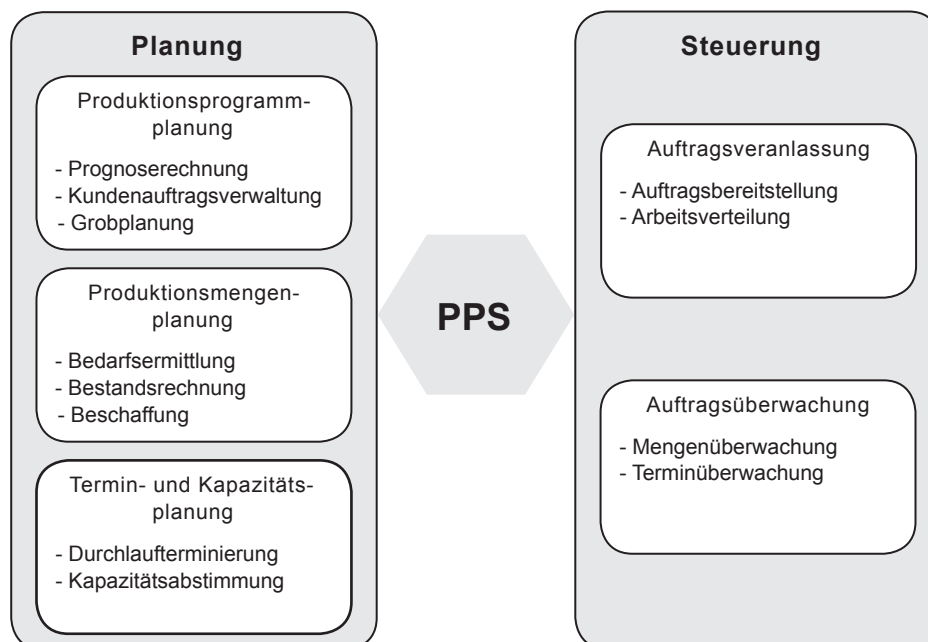


Abbildung 3: Funktionen der Produktionsplanung und -steuerung [Ze06, S.16]

2.3 Die Ablaufsimulation im fertigungstechnischen Umfeld

2.3.1 Begriffsbestimmung und Definition

Gemäß der VDI-Richtlinie 3633 bezeichnet **Simulation**⁴:

„[...] das Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind. [...] Im weiteren Sinne wird unter Simulation das Vorbereiten, Durchführen und Auswerten gezielter Experimente mit einem Simulationsmodell verstanden“ [VDI 3633-1, S.2].

Nachfolgend werden die wesentlichen Begriffe dieser Definition eingegrenzt.

„Ein **System** ist eine abgegrenzte Anordnung von Komponenten, die miteinander in Beziehung stehen“ (vgl. ebenda S. 3). Gekennzeichnet ist es durch (vgl. ebenda S. 3):

- Eine Grenze gegenüber seiner Umwelt, die Systemgrenze. Über Schnittstellen kann ein Austausch von Materie, Energie und Informationen stattfinden.
- Komponenten, die, falls sie nicht weiter zerlegbar sind, als „Systemelemente“ bezeichnet werden. Oder bei Erhöhung des Detaillierungsgrades selbst wiederum Systeme darstellen. Sie werden Subsysteme genannt.
- Die Ablaufstruktur innerhalb der Komponenten, welche durch Regeln und konstante oder variable Attribute festgelegt wird.
- Relationen, welche die Systemkomponenten miteinander verbinden, um den Ablauf eines Prozesses zu ermöglichen.
- Zustände der Komponenten, die durch die Angabe der Werte aller konstanten und variablen Attribute beschrieben werden.
- Zustandsübergänge der Komponenten. Diese stellen kontinuierliche oder diskrete Änderungen mindestens einer Zustandsgröße dar. Die Änderungen von Zustandsgrößen werden durch den im System ablaufenden Prozess hervorgerufen.

„Ein **Modell** ist eine vereinfachte Nachbildung eines geplanten oder existierenden Systems mit seinen Prozessen in einem anderen begrifflichen oder gegenständlichen System. Es unterscheidet sich hinsichtlich der untersuchungsrelevanten Eigenschaften nur innerhalb eines vom Untersuchungsziel abhängigen Toleranzrahmens vom Vorbild“ (vgl. ebenda S. 3).

4 Im Rahmen dieser Arbeit soll unter dem Begriff Simulation die Ablauf- bzw. Materialflusssimulation verstanden werden.

„Ein **Simulationsexperiment** ist die gezielte empirische Untersuchung des Verhaltens eines Modells durch wiederholte Simulationsläufe mit systematischer Parametervariation oder Strukturvariation“ (vgl. ebenda S. 3).

Im fertigungstechnischen Umfeld existieren eine Reihe von Einsatzgebieten der Simulationstechnik. Diese reichen von sehr detaillierten Untersuchungen der Produktkonstruktion mittels Finite-Elemente-Simulation oder Betrachtungen einzelner Maschinen bzw. Maschinenzellen bis hin zu stark aggregierten, aber wenig detaillierten Geschäftsprozesssimulationen zur Unterstützung strategischer Unternehmensentscheidungen [Fr07, S.37f]. In dieser Arbeit wird die Ablaufsimulation im Fokus der Betrachtungen stehen. Sie wird für die Abbildung von Material- und Informationsflüssen geplanter oder bestehender Fertigungssysteme eingesetzt. Klassische Zielstellungen bestehen z. B. darin, die Grenzleistung eines Systems zu ermitteln, den Einfluss festgelegter Systemparameter zu bestimmen und Strategien bei der Störfallbeseitigung zu bewerten [Fr03, S.15f]. Ein Großteil der Ablaufsimulationen basiert auf ereignisorientierten Modellen [Ra03, S.7]. Bei der ereignisorientierten Simulation werden nur Zeitpunkte betrachtet, zu denen eine Änderung des Systemzustandes stattfindet. Die im fünften Kapitel vorgestellte automatisierte Simulationsmodellgenerierung basiert auf dem ereignisorientierten Paradigma.

2.3.2 Anwendungsgebiete der Ablaufsimulation

Die Ablaufsimulation findet in allen Phasen des Lebenszyklus von technologischen Systemen Anwendung. Ihr Einsatz erstreckt sich von der Planung über die Realisierung bis hin zum Betrieb des Systems. Hierbei können folgende typische Anwendungsfelder der Simulation zusammengefasst werden [VDI 3633-1, S.4f].

In der Planungsphase:

- Überprüfung geplanter Anlagenkonzepte mit dem Ziel des Funktionsnachweises, der Leistungserbringung, Effizienzsteigerung und Projektkostenminimierung hinsichtlich Anlagendimensionierung, Ablauflogik, Steuerungskriterien, Durchsatz, Leistungsgrenzen, Engpässen, Durchlaufgeschwindigkeit, Bestandsbildung
- Verbesserung/Optimierung existierender Anlagen durch Ermittlung von Kapazitätsgrenzen und Schwachstellen, Beurteilung der Auswirkungen von Veränderungen der Kapazitäten, Produkten, Produktionsplänen, Systemstrukturen, Organisationsstrukturen und Führungsstrategien

In der Realisierungsphase:

- Leistungstests der Anlage bei schrittweiser Einsteuerung der Auftragsstypen und Produktvarianten sowie bei schrittweiser Kapazitätsauslastung
- Überprüfung von Auswirkungen von Anforderungsänderungen während der Installation der Anlagen
- Erprobung und Tests der Steuerungssoftware
- Mitarbeiterschulung bei der Behandlung von Notfällen und Störungen

In der Betriebsphase:

- Variantenuntersuchungen zur operativen Entscheidungsunterstützung für Maschinenbelegungen, Auftragsreihenfolgen, Losgrößen und Personaleinsatz
- Überprüfen von Notfallstrategien bezüglich ihrer Auswirkungen auf z. B. die Ausbringungsmenge, Durchlaufzeiten und Bestände
- Variantenuntersuchungen für Prognosezwecke hinsichtlich veränderter Produktstruktur, Produktmix, Ausbringungsmenge, veränderter Arbeitszeitmodelle, Maschinenparks und Arbeitsabläufe
- Unterstützung im Bereich der Produktionsplanung und -steuerung (PPS) z. B. innerhalb der Mengenplanung zur genaueren Bedarfsermittlung des Materials (Mengen und Termine) oder zur simultanen Einplanung knapper Ressourcen (Mensch, Maschine, Werkzeug, Material) bei der Termin- und Kapazitätsplanung

PPS-Simulation:

Die Anwendung der Simulationstechnik im Bereich der PPS ist aus der Intension heraus entstanden, die Schwachstellen herkömmlicher PPS-Systeme zu überwinden. Eckardt führt die Schwachstellenpunkte klassischer PPS-Systeme auf die „fehlende Berücksichtigung einer *integrierten, simultanen* Material-, Personal-, und Betriebsmittelplanung unter zeitdynamischen Aspekten“ [Ec02, S.11] zurück. Die Schwachstellen führen in taktischen und operativen Planungszeiträumen überwiegend zu unbrauchbaren Plänen (vgl. ebenda S.11). Hauptkritikpunkt an konventionellen PPS-Systemen ist die Sukzessivität der Planung, bei der die einzelnen Planungsschritte sequentiell durchlaufen werden, wodurch eine Vernachlässigung der gegenseitigen Abhängigkeiten der Planungsstufen erfolgt [Ec02, S.9].

Die Simulation bietet in den folgenden Aufgabengebieten der PPS Unterstützung [VDI 3633-5, S.3f]:

- Produktionsprogrammplanung

Sie umfasst die Prognoserechnung, Grobplanung, Lieferterminbestimmung, Kundenauftragsverwaltung und Vorlaufsteuerung. Die Simulation kann dazu beitragen, die kapazitiven und materialflusseitigen Auswirkungen von Änderungen im Produktionsprogramm aufzuzeigen.

- Produktionsmengenplanung

Ihre Aufgaben sind die Bedarfsermittlung, Bestandsführung und die Beschaffung (Eigenerfertigung oder externe Bestellaufträge). Klassische PPS-Verfahren ermitteln die Materialbedarfstermine mit fixen Vorlaufzeiten, was in der Praxis mit dem Ziel der Reduzierung der Bestände zu ungenau sein kann. Mithilfe der Abbildung dynamischen Verhaltens in Simulationsmodellen können die erforderlichen Mengen und Termine wesentlich genauer bestimmt werden.

- Termin- und Kapazitätsplanung

In diesem Bereich findet die Durchlaufzeitermittlung, Kapazitätsbedarfsermittlung, Kapazitätsbestimmung und Reihenfolgeplanung statt. Die Simulation ermöglicht eine gleichzeitige und gleichberechtigte Einplanung mehrerer Ressourcen und die Berücksichtigung begrenzter Kapazitäten als Prüfkriterium der Entscheidung. Zudem können mehrstufige Produktions- und Logistikprozesse mit mehrfacher Zielsetzung gemeistert werden, da die Simulation vorab realitätsnahe Auskünfte zur Zielerreichung in Bezug auf z. B. Termintreue, Durchlaufzeit, Kapazitätsauslastung und Materialbestände geben kann.

- Auftragsfreigabe

Sie umfasst die Freigabe der Fertigungsaufträge, die Arbeitsverteilung und die Bestellauftragsfreigabe. Die Auftragsfreigabe kann mithilfe der Simulation durch eine dynamische Verfügbarkeitsprüfung unterstützt werden, wodurch eine Verringerung der Durchlaufzeiten gegenüber konventionellen PPS-Verfahren erreicht werden kann. So werden vornehmlich Aufträge freigegeben, deren benötigtes Material zum realen Fertigungszeitpunkt auch zur Verfügung stehen wird.

- Auftragsüberwachung

Die Aufgaben der Auftragsüberwachung sind die Auftragsfortschritts-, Kapazitäts- und Bestellauftragsüberwachung sowie die Wareneingangserfassung. Ist das Simulationsmodell mit dem aktuellen Ist-Zustand des realen Fertigungssystems synchronisiert, können Auswirkungen von Störungen im Betriebsablauf aufgezeigt und frühzeitig Maßnahmen ergriffen werden, um die Zielerreichung (z. B. Termintreue) nicht zu gefährden.

Die Überwindung der Schwächen klassischer PPS-Systeme mithilfe der Simulation zielt darauf ab, einen Regelkreis zwischen Produktionsplanung und -steuerung und dem realen Fertigungssystem mittels der Simulation zu etablieren. Das Simulationssystem ist dabei in der Lage, zukünftiges Prozessgeschehen vorwegzunehmen und erwartete Ergebnisse der PPS zur Verfügung zu stellen (vgl. Abbildung 4).

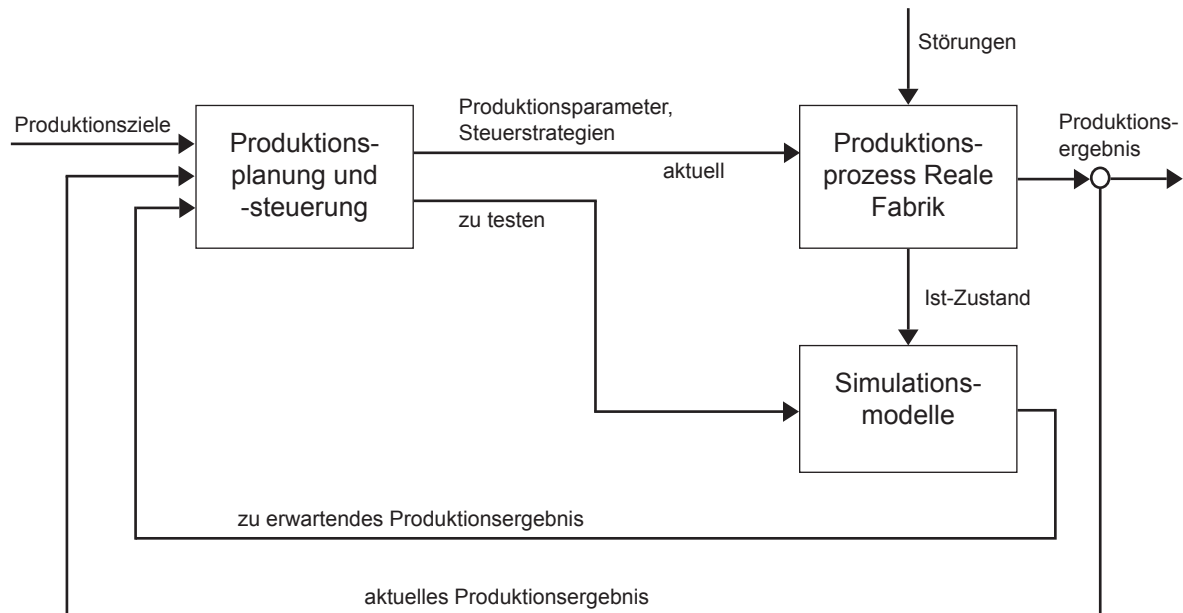


Abbildung 4: Einsatz der Simulation zur Unterstützung der operativen Produktionsplanung [Kü06, S.90]

Eine von Reinhart und Feldmann durchgeführte Studie zum Einsatz der Simulationstechnik im fertigungstechnischen Umfeld in Deutschland erbrachte die in Abbildung 5 festgehaltene Verteilung der Einsatzfelder der Ablaufsimulation. Die Fertigungsplanung beinhaltet die Anwendungen innerhalb der Phasen Planung und Realisierung.

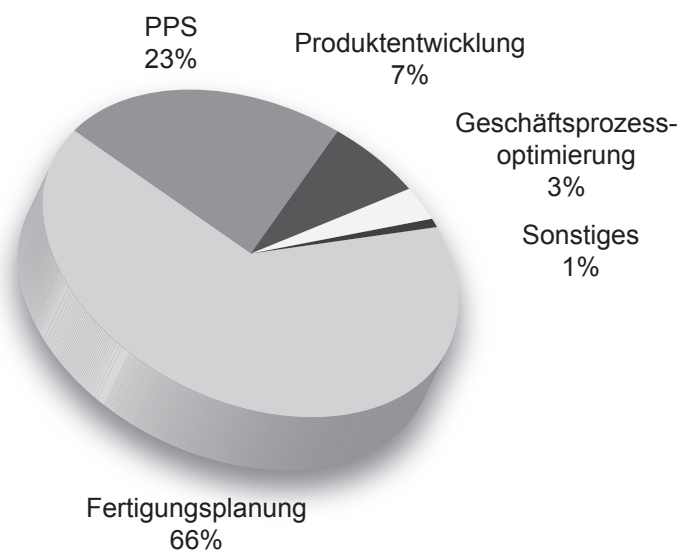


Abbildung 5: Anwendungsgebiete der Ablaufsimulation nach [ReFe97a, S.19]

Das dominierende Einsatzgebiet ist demnach die Fertigungsplanung, und innerhalb dieser überwiegt die Anwendung in der Anlagenplanung. Eingesetzt wird die Simulation hierbei zur Optimierung des Layouts, Dimensionierung der Anlage sowie zur Überprüfung und Absicherung des Anlagenkonzeptes.

Neben dem Einsatz bei der Planung von Produktionseinrichtungen hat sich die Anwendung für die fertigungsbegleitende Produktionsplanung und -steuerung (PPS) etabliert. Schwerpunkte liegen dabei auf der Kapazitätsplanung, Optimierung von Einlaststrategien, Fertigungssteuerung und der Anlagenüberwachung [ReFe97a, S.18f].

2.3.3 Nutzen der Simulation

Eine genaue Quantifizierung der Nutzenaspekte des Simulationseinsatzes ist zu Beginn des Projektes meist nur sehr schwer oder gar nicht möglich [Kü06, S.23]. Beim planungsbegleitenden Simulationseinsatz lässt sich kaum nachvollziehen, wann, wo und in welchem Maße die Einsparungen auf die Anwendung der Simulation zurückgeführt werden können. Denn bereits bei der Erstellung von Konzepten leistet die Simulation ihren Beitrag zum besseren Systemverständnis und trägt dazu bei, dass neue Erkenntnisse in die Planung miteinfließen können [VDI 3633-1, S.21]. Dennoch können gemäß VDI-Richtlinie 3633 folgende Nutzenaspekte des Simulationseinsatzes konstatiert werden [VDI 3633-1, S.22]:

„Sicherheitsgewinn:

- Bestätigung der Planungsvorhaben (bei positiver Bewertung; Anm. d. Verf.)
- Minimierung des unternehmerischen Risikos
- Funktionalität des geplanten Systems
- Funktionalität der Steuerung
- Qualität des Pflichtenheftes

Kostengünstigere Lösungen:

- Einsparung oder Vereinfachung von Systemelementen
- Einsparung oder Vereinfachung von Steuerungselementen
- Optimierung von Puffergrößen und Lagerbeständen
- Optimierung von Arbeitsabläufen(-inhalten)

Besseres Systemverständnis:

- Parametersensitivitäten
- Begründbarkeit und Überprüfung der gewählten Lösung
- Vermeidung oder Eliminierung von Engpässen
- Schulung des Betriebspersonals
- dynamische Analyse und Darstellung des gesamten Ablaufs (Animation)

Günstigere Prozeßführung:

- Entscheidungsunterstützung bei Betriebsproblemen
- Prozeßoptimierung nach beliebigen Zielfunktionen (z.B. Durchlaufzeit, Auslastung, Ausbringung)
- Produktivitätssteigerung
- Optimierung der Anlagensteuerung
- Minimierung der im Störfall entstehenden Ausfallkosten
- Verkürzung der Anlaufphase“

2.3.4 Phasen einer Simulationsstudie

Bei der Durchführung von Simulationsstudien gliedert sich die Vorgehensweise nach VDI-Richtlinie 3633 in die drei Phasen: Vorbereitung, Durchführung und Auswertung [VDI 3633-1, S.10f]. Einen Überblick hierzu gewährt Abbildung 6.

Zu Beginn einer Simulationsstudie ist zunächst einmal die Grundsatzentscheidung zu treffen, ob die vorliegende Problemstellung überhaupt simulationswürdig ist.

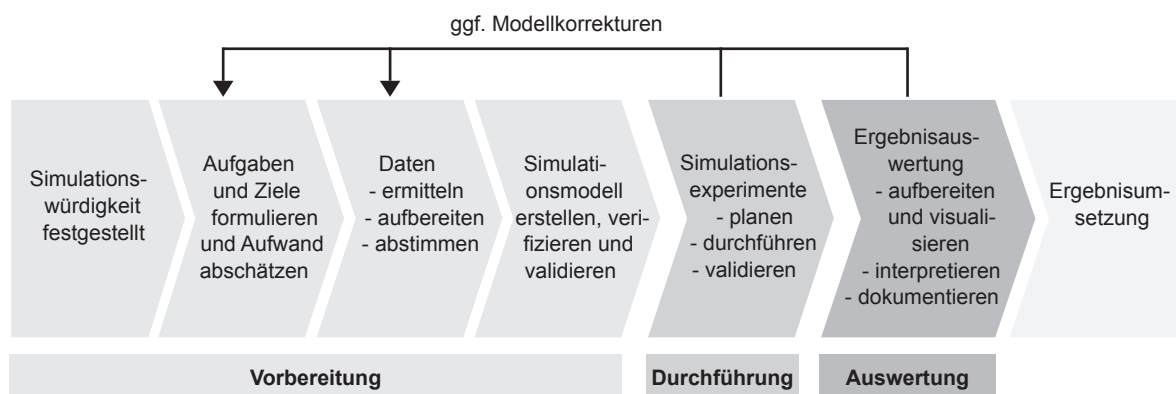


Abbildung 6: Phasen einer Simulationsstudie nach [VDI3633-1, S.10f; Ro09, S.48]

Die VDI-Richtlinie 3633 benennt zur Unterstützung folgende Kriterien (vgl. ebenda S. 12):

- „Günstiges Kosten-Nutzen-Verhältnis
- Schwierigkeit des Problems (Komplexität, Unüberschaubarkeit, Vielzahl der Einflüsse, Abhängigkeiten und nebenläufige Prozesse, große Datenmengen)
- Unsicherheit bezüglich der Daten oder der mathematischen Verteilung streuender Werte und ihres Einflusses auf die Ergebnisgrößen
- Fehlen analytischer mathematischer Modelle bzw. ungerechtfertigt hoher Aufwand (Zeit/Kosten)
- Ungleichgewicht zwischen vorliegender Datenqualität und erwarteter Modellqualität

- Sicherheitsbedürfnis bei unscharfen Vorgaben, Ausloten von Grenzen, Beweisnot
- wiederholte Verwendung des Simulationsmodells“

Nachdem die Simulationswürdigkeit festgestellt wurde, sind Aufgabenstellungen und Ziele der Untersuchungen zu definieren. Hierbei ist es wichtig, Kenngrößen festzulegen, die das Verhalten des zu untersuchenden Systems bezüglich seiner Aufgabe charakterisieren und durch die Simulation ermittelbar sind. So können z. B. die Auftragsdurchlaufzeit, die Maschinenauslastung, die Höhe der Bestände von Teilen und Zwischenprodukten sowie die Terminabweichung Kenngrößen sein, um die Leistungsfähigkeit und Wirtschaftlichkeit eines Fertigungssystems zu beurteilen (vgl. ebenda S.12). Eine umfassende Analyse des zu betrachtenden Systems mündet schließlich in der Festlegung der Modellstruktur, der Systemgrenzen und des Detaillierungsgrades des benötigten Simulationsmodells [Ro09, S.48]. Anschließend werden für das Simulationsmodell relevante Daten ermittelt und aufbereitet. In einem weiteren Schritt erfolgt die Erstellung eines experimentierfähigen Modells eines bereits existierenden oder gedachten Systems. Ob die Modellspezifikation richtig in das Simulationsmodell überführt wurde, wird im Rahmen der Verifikation überprüft. Eine erfolgreiche Validierung stellt sicher, dass Original und Modell in hinreichendem Maße übereinstimmen. Die Planung und Durchführung der Simulationsexperimente findet in der nun anschließenden Phase statt. Schließlich werden in der letzten Phase die gewonnenen Ergebnisse ausgewertet und geeignet aufbereitet, um eine Ergebnisinterpretation zu ermöglichen, sowie aussagekräftig zusammengefasst und ggf. unter Zuhilfenahme von Animationen visualisiert und schlussendlich dokumentiert.

2.3.5 Eingangsdaten für die Simulation

„Die Menge, Art und Anzahl der benötigten Daten zur Durchführung von Simulationsstudien variiert stark mit der Zielsetzung, dem Detaillierungsgrad und der Methode der durchzuführenden Simulationsstudie“ [Je07, S.45]. Dennoch lassen sich die Eingangsdaten nach VDI-Richtlinie 3633 in die Kategorien technische Daten, Organisationsdaten und Systemlastdaten einordnen [VDI 3633-1, S.13] (vgl. Abbildung 7).

Zusätzlich soll in Analogie zu Selke [Se05, S.15f] und Splanemann [Sp95, S.43] die Kategorie der Experimentdaten unterschieden werden. Daten dieser Klasse dienen nicht der Charakterisierung des Simulationsmodells, sondern sind für die Durchführung der Simulationsexperimente von großer Bedeutung. Hierzu zählen u. a. die Einschwingdauer, die Versuchsanzahl und die Versuchsdauer. Diese werden unter Zuhilfenahme von statistischen Methoden vor Beginn der Experimentierphase ermittelt. Hierzu ist eine detaillierte Beschreibung in [Fr07] zu finden.

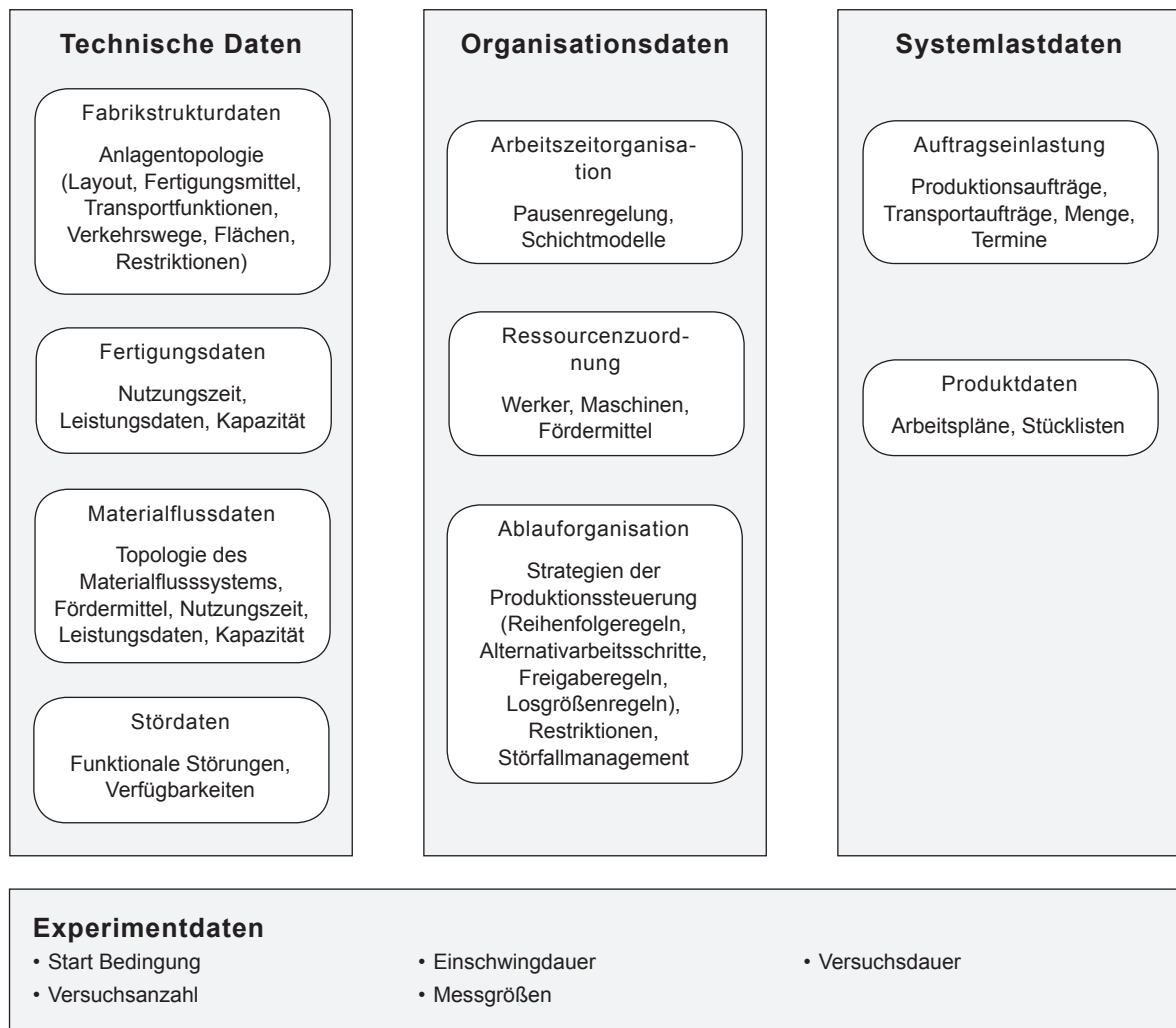


Abbildung 7: Gruppierung der Eingangsdaten für Ablaufsimulationen nach [VDI 3633-1, S.12f; Se05, S.15f]

Zur Kategorie der technischen Daten gehören Angaben zur Fabrikstruktur, wie z. B. das Anlagenlayout, Logistikflächen sowie Transportwege, aber auch Leistungs- und Kapazitätswerte und Angaben zum Störungsverhalten fertigungstechnischer Anlagen.

Den Organisationsdaten zugehörig sind Angaben zu Schichtmodellen, der Zuordnung von Werkern und Arbeitsstationen und die Strategien der Produktionssteuerung. Unter den Systemlastdaten werden Daten zu Stücklisten, Arbeitsplänen sowie auftragsbezogene Daten, wie Fertigungs- und Transportaufträge zusammengefasst.

Abläufe und Strategien der Produktionssteuerung:

Unter dem Begriff Strategie soll eine prinzipielle Vorgehensweise zur Erreichung eines definierten Zieles verstanden werden [ReFe97b; Se05, S.41].

„Ein Ablauf ist ein konkretes Vorgehen, das unter gegebenen Randbedingungen das Umsetzen einer übergeordneten Strategie und somit das Erreichen des Ziels erlaubt. Ein Ablauf er-

zeugt eine Zustandsänderung, in der Realität genauso wie im Simulationsmodell [...]“ [Se05, S.41] nach [ReFe97b].

Dass Fertigungen in aller Regel nach dem Wirtschaftlichkeitsprinzip gestaltet werden, wurde in Abschnitt 2.2 dargelegt. Von zentraler Bedeutung ist demnach die Fertigung an wirtschaftlichen Ersatzzielgrößen, wie kurze Durchlaufzeiten, hohe Termintreue, hohe Auslastung und geringe Bestände, auszurichten. Strategien und Abläufe der Produktionssteuerung beziehen sich auf diese Zielgrößen und sollen eine möglichst gute Erreichung dieser sicherstellen. Eine mögliche Clustereinteilung der Strategien und Abläufe der Produktionssteuerung ist Tabelle 1 zu entnehmen.

Strategiecluster	Beispiele
Strategien zur Losgrößengestaltung	<ul style="list-style-type: none"> • Splittung eines Loses und parallele Bearbeitung auf mehreren Maschinen • Zusammenfassen von Losen aus technologischen Gründen (z. B. Ofenkapazität bei der Wärmebehandlung) • Überlappung: Teilmengen eines Loses werden zur nächsten Arbeitsstation gebracht, obwohl das Los noch nicht fertiggestellt ist
Strategien zur Ressourcenbelegung (Auswahl und Terminierung)	<p>Zuteilung der Ressource, die:</p> <ul style="list-style-type: none"> • am geringsten ausgelastet ist • den geringsten Rüstaufwand bietet • am schnellsten verfügbar ist
Strategien zur Reihenfolgegestaltung	<p>Höchste Priorität in einer Warteschlange vor einem Arbeitsplatz hat der Auftrag</p> <ul style="list-style-type: none"> • der die Warteschlange zuerst erreicht (FIFO) • mit der kürzesten Bearbeitungszeit (KOZ) • mit der kürzesten Restbearbeitungszeit (KRB)
Strategien zur Auftragsfreigabe	

Tabelle 1: Einteilung der Strategien und Abläufe der Produktionssteuerung nach [Se05, S.45ff]

In der Praxis werden einzelne Strategien oft modifiziert oder durch Kombination mehrerer miteinander verbunden. Hierdurch kann prinzipiell eine unendliche Anzahl an Strategien und Abläufen entstehen [Se05, S.45].

Möglichkeiten der Nutzung stochastischer Daten:

Fertigungssysteme unterliegen in aller Regel stochastischen Einflussgrößen, d. h. der Folgezustand des Systems ist nicht exakt vorhersehbar, sondern wird zufallsabhängig bestimmt. So sind z. B. das Auftreten von Maschinenstörungen, die dafür benötigten Reparaturzeiten und die Ankunftszeiten von Fertigungsaufträgen zufallsabhängige Größen. Reicht die Verwendung von Mittelwerten bei der Systembetrachtung nicht aus, können unter Verwendung historischer Datensätze stochastische Einflüsse miteinbezogen werden. Hierbei sind zur Übernahme in das Simulationsmodell prinzipiell drei Ansätze möglich [Kü06, S.152f] (vgl. Abbildung 8):

- „direkte Nutzung der erfassten Daten
- Anpassung einer stochastischen Verteilung an die erfassten Daten
- Generieren einer empirischen Verteilung aus den erfassten Daten“

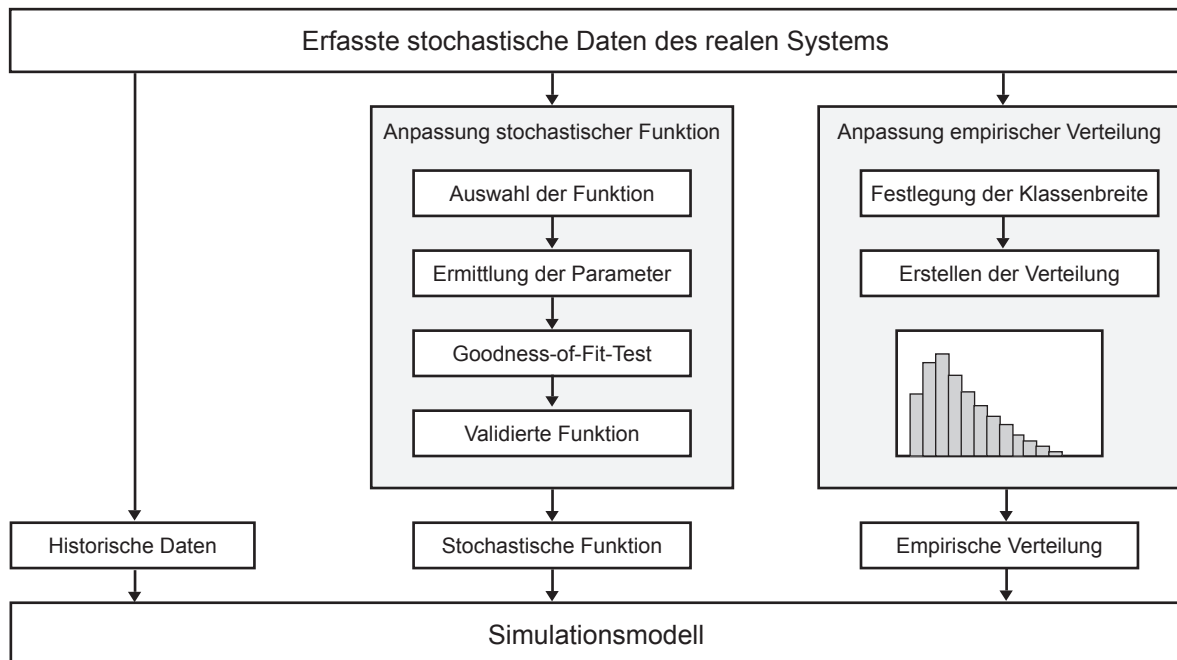


Abbildung 8: Möglichkeiten der Nutzung stochastischer Daten für die Simulation [Kü06, S.153]

Während die direkte Nutzung historischer Daten für die Modellvalidierung hilfreich sein kann, wird in den beiden anderen Fällen eine kompaktere Beschreibung erzeugt. Diese verbessert die Handhabbarkeit und ermöglicht die eventuelle Übertragbarkeit auf noch nicht existierende Anlagenteile (vgl. ebenda S. 152).

Bei der Anpassung stochastischer Funktionen ist es in der Regel notwendig, die entsprechenden Verteilungsparameter zu schätzen und anschließend mit „Goodness-of-Fit“ Tests die Güte der modifizierten Verteilung zu überprüfen. Alternativ können die erfassten stochastischen Daten auch mithilfe empirischer Verteilungen verdichtet werden. Hierfür sind lediglich die Intervallgrenzen und -häufigkeiten auszuwählen (vgl. ebenda S. 153).

2.4 Die Digitale Fabrik

2.4.1 Begriffsbestimmung und Definition

Nach der VDI-Richtlinie 4499 wird die Digitale Fabrik wie folgt definiert [VDI 4499, S.3]:

„Die Digitale Fabrik ist der Oberbegriff für ein umfassendes Netzwerk von digitalen Modellen, Methoden und Werkzeugen – u. a. der Simulation und der drei-

dimensionalen Visualisierung – , die durch ein durchgängiges Datenmanagement integriert werden.

Ihr Ziel ist die ganzheitliche Planung, Evaluierung und laufende Verbesserung aller wesentlichen Strukturen, Prozesse und Ressourcen der realen Fabrik in Verbindung mit dem Produkt. “

Die Digitale Fabrik bildet die Grundlage für die integrierte Zusammenarbeit zwischen Produktentwicklung und Fertigungsplanung, da sie die simultane Entwicklung von Produkt und Fertigung unterstützt [Ro09, S.6]. Aufgabe ist es, Produkte, Fertigungsverfahren und Fertigungsabläufe bereits in frühen Phasen der Entwicklung abzusichern sowie die Fabrik- und Fertigungsplanung mit rechnergestützten Modellen und Werkzeugen zu begleiten und zu beschleunigen. Weiterhin soll die reale Fertigung fortlaufend durch den Einsatz der Methoden und Werkzeuge der Digitalen Fabrik überprüft und verbessert werden [VDI 4499, S.4]. Der Fokus der Digitalen Fabrik liegt im Bereich der Fabrik- und Fertigungsplanung, jedoch erstreckt sich ihr Wirkungsbereich von der Produktentwicklung bis hin zur realen Fertigung (vgl. Abbildung 9).

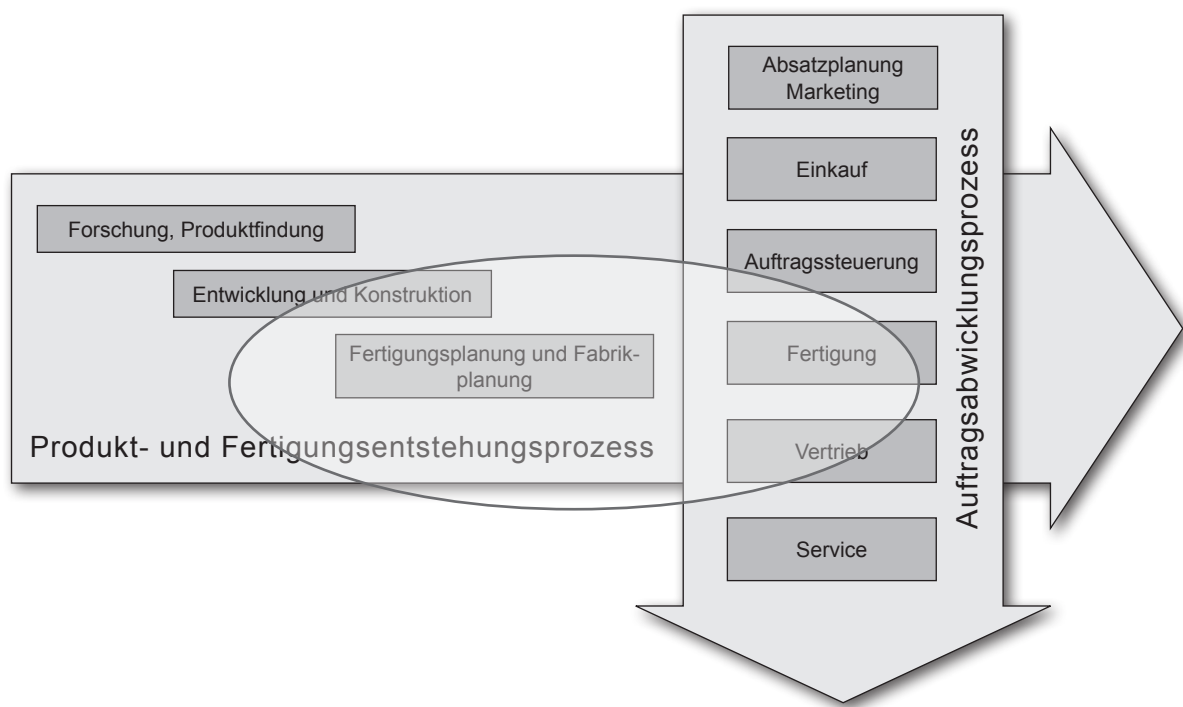


Abbildung 9: Fokus der Digitalen Fabrik nach [VDI 4499, S.3]

Die Technologie der Digitalen Fabrik ermöglicht es, „in einer virtuellen Fabrik⁵ die Produkte, Prozesse und Anlagen in Modellen abzubilden und basierend auf den virtuellen Produkten und Prozessen die geplante Fertigung virtuell am Rech-

⁵ Unter dem Begriff „virtuelle Fabrik“ soll ein modellhaftes Abbild einer existierenden oder geplanten Fabrik verstanden werden, deren Modelle alle planungsrelevanten Informationen über die Produkte, Prozesse und Ressourcen enthalten und rechnergestützt verarbeitet werden können.

ner so zu verbessern, dass ein ausgereifter Prozess für die reale Fabrik zur Verfügung steht“ [Kü06, S.2].

Trotz des scheinbaren Schwerpunktes der Digitalen Fabrik auf Softwarewerkzeuge, kann sie nicht auf den Bereich der Informationstechnologie reduziert werden. Vielmehr vereint sie die rechnergestützte Planung mittels geeigneter Software mit dem konsequenten und durchgängigen Einsatz von Planungsmethoden. Bei der Umsetzung der Digitalen Fabrik muss gewährleistet sein, dass sie im Einklang mit der Einführung und Entwicklung der Planungsmethoden, deren Einsatz rechnergestützt erfolgen muss, stattfindet [Fr07, S.12; BFZ06, S.20].

2.4.2 Komponenten der Digitalen Fabrik

Die Digitale Fabrik gliedert sich in die folgenden vier Komponenten [Ro09, S.7] (vgl. Abbildung 10):

- Eine strukturierte Datenbasis als Datengrundlage
- Eine Integrationsplattform als Schnittstelle zwischen Daten und Werkzeugen
- Rechnergestützte Werkzeuge, die über Schnittstellen angebunden sind
- Eine übergeordnete Organisation und ein übergreifender Planungsablauf

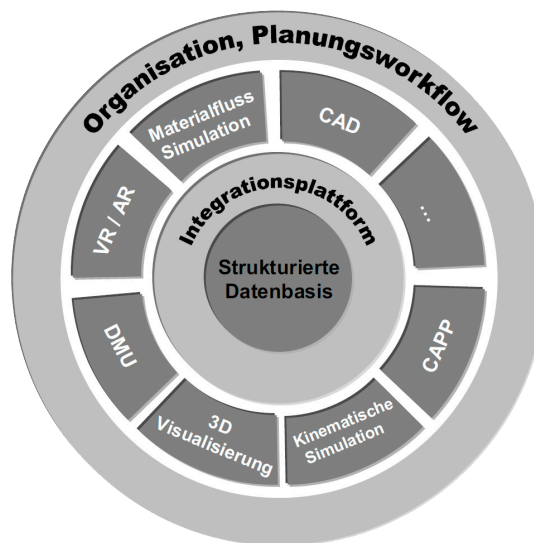


Abbildung 10: Struktur der Digitalen Fabrik nach [BRK04, S.28] aus [Ro09, S.7]

Den Kern der Digitalen Fabrik bildet eine Datenbasis, in welcher sämtliche relevante Daten zu Produkten, Ressourcen, Vorgängen und des Layouts strukturiert abgelegt und verwaltet werden. Ein durchgängiges Datenmanagement gewährleistet den Zugriff der Softwarewerkzeuge auf die Daten und stellt deren konsistente Speicherung sicher. Der Integration kommt die Aufgabe zu, „die unterschiedlichen Systeme zu einem harmonischen Ganzen zusammenzuführen und in die bestehende IT-Infrastruktur zu integrieren“ [Ro09, S.10].

Abbildung 11 zeigt exemplarisch die Bandbreite der Anwendungsfelder, in denen rechnergestützte Systeme zum Einsatz kommen.

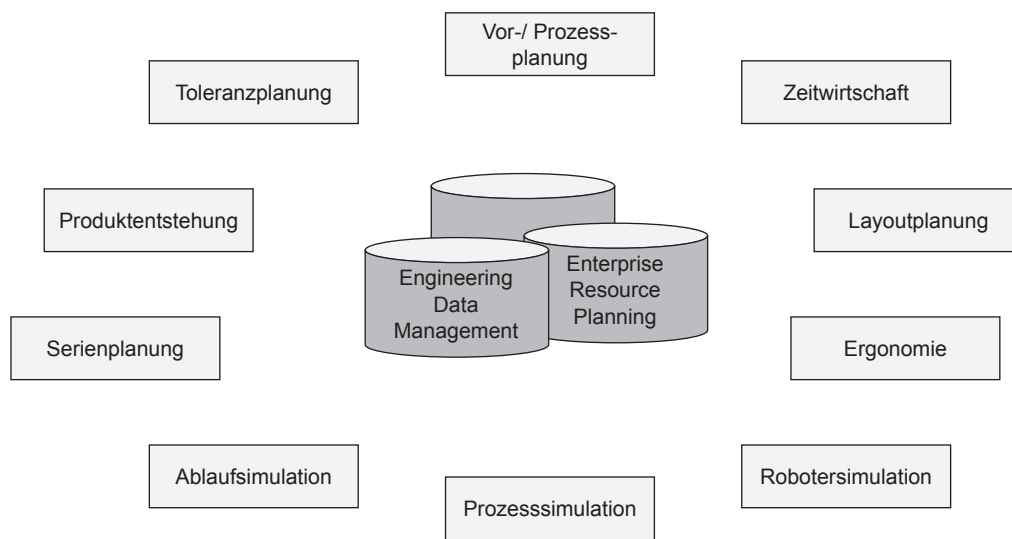


Abbildung 11: Anwendungsfelder für Werkzeuge der Digitalen Fabrik nach [BäHa03, S.152]

Diese reichen von der Konstruktion der Produkte über die Prozess- und Layoutplanung, anschließenden Ergonomiebetrachtungen, Roboter- sowie Ablaufsimulationen einzelner Fertigungszellen oder ganzer Produktionslinien einschließlich logistischer Abläufe bis hin zur Serienfertigung.

2.4.3 Ablaufsimulation in der Digitalen Fabrik

Die Ablaufsimulation, als ein wichtiger Bestandteil der Digitalen Fabrik, wird vorwiegend auf Anlagen- und Zellenebene eingesetzt (vgl. Tabelle 2). Ihr Aufgabenbereich liegt hauptsächlich in der dynamischen Bewertung von Anlagenlayouts, Logistik- und Produktionsflüssen. Wichtige Ziele der Untersuchungen sind hierbei [Kü06, S.60]:

- Analyse und Optimierung von Fabrik- und Anlagenlayouts
- Frühzeitige dynamische Überprüfung des Fertigssystems im Hinblick auf Leistungsparameter (z. B. Durchlaufzeiten und Ausbringungsmenge)
- Ermittlung von Engpässen
- Optimierung von Lagerflächen und Transportwegen
- Dynamische Überprüfung der Pufferauslegung

Weitere Anwendungsgebiete sind die Unterstützung in der Inbetriebnahmephase des Fertigungssystems, bei der die Ablaufsimulation das reale System ersetzt und für die Steuerungssoftware als Emulator fungiert, wie auch der Einsatz der Simulation zur kontinuierlichen

Verbesserung der Prozessabläufe und der Struktur im Sinne eines kontinuierlichen Redesign der Fabrik [Kü06, S.86 u. 93].

Ebene	Planungsinhalte	Simulationstechnik
Unternehmen	Geschäftsprozesse Informationsflüsse Fertigungsstrategien	Geschäftsprozesssimulation
Fertigungssystem	Anlagenlayout Materialflusslogistik Steuerungsstrategien Arbeitsorganisation	Ablaufsimulation
Zelle	Zellenlayout RC-Programmierung NC-Programmierung Taktzeitoptimierung Kollisionsvermeidung	Grafische 3D-Simulation Kinematiksimulation Ablaufsimulation
Komponente	Operationen Prozessparameter Werkzeuge Hilfsmittel	Mehrkörpersimulation FEM-Simulation

Tabelle 2: Anwendung der Simulationstechnik in verschiedenen Bereichen produzierender Unternehmen nach [Kü06, S.56]

2.4.4 Ziele der Digitalen Fabrik

Die mit der Digitalen Fabrik angestrebten Ziele sind [Kü06, S.5]:

- „Verbesserung der Wirtschaftlichkeit
- Verbesserung der Planungsqualität
- Verkürzung der Produkteinführungszeit
- Transparente Kommunikation
- Standardisierung von Planungsprozessen
- Kompetentes Wissensmanagement“

Erhebliche Nutzenpotenziale werden in der Integration von Produktentwicklung und Fertigungsplanung gesehen. Hierdurch soll eine Beschleunigung der Planungs- und Inbetriebnahmeprozesse erreicht sowie die Qualität der Produkte und Planungsprozesse verbessert werden [Kü06, S.6]. Als Maßnahmen zur Erreichung dieser Ziele werden eine weitgehende Parallelisierung der Einzelprozesse auf Basis gemeinsam genutzter rechnergestützter Modelle, die Nutzung redundanzfreier, aktueller und konsistenter Daten und die Verbesserung der Zusammenarbeit von Entwicklung, Planung, Produktion und Lieferanten angestrebt [Kü06, S.6; VDI 4499, S.9]. Eine Verbesserung der Planungsqualität kann durch die Integration der Fertigungsplanung erreicht werden. Die Methoden und Werkzeuge der Digitalen Fabrik bie-

ten Unterstützung bei Aktualität und Transparenz der Planungsgrundlagen. Überdies wird die Erreichung eines durchgängigen Planungsprozesses in Bezug auf Daten, Informationsmanagement und Ressourcen gefördert. Zu einer erhöhten Planungssicherheit trägt hierbei die enge Verzahnung von Produktentwicklung und Fertigungsplanung bei [Kü06, S.8; VDI 4499, S.11]. So kann mithilfe geometriebasierter Planung auf Basis der Produktdaten und Ressourcen (z. B. durch rechnergestützte 3D-Prozess- und 3D-Montageplanung) das frühzeitige Erkennen von Konstruktions- und Prozessfehlern erleichtert werden. Dies hilft Verzögerungen des Fertigungsanlaufs zu verhindern und leistet einen wesentlichen Beitrag zur Reduzierung der Änderungskosten [Ro09, S.17].

Weiterhin sollen Planungsprozesse optimiert und die Effizienz gesteigert werden. Dies ist insbesondere deshalb von Bedeutung, da die Anforderungen an die Planungstätigkeiten in den letzten Jahren gestiegen sind, z. B. indem Planungszeiträume verkürzt werden und eine höhere Flexibilität des Fertigungssystems erwartet wird. Welche Potenziale zur Verbesserung noch ungenutzt sein können, zeigt das Beispiel einer Analyse der CIM Aachen. So wurde herausgefunden, dass die Fertigungsplaner in dem untersuchten Unternehmen die Hälfte ihrer Arbeitszeit mit der Dokumentation und Informationsbeschaffung verbrachten. Für die eigentliche kreative Planungstätigkeit wurde hingegen nur ein Fünftel der Arbeitszeit aufgewendet [Ma05, S.40].

Rooks beschreibt vier ineinandergreifende Stoßrichtungen der Fertigungsplanung, um Effizienzsteigerungen der Planung durch die Digitale Fabrik zu erzielen: Standardisierung, Integration, Automatisierung und Workflow [Ro09, S.15f]. Herausgestellt seien hiervon außer der bereits erwähnten Integration, die Standardisierung und Automatisierung. Eine Standardisierung erfolgt bezüglich der Produkt- und Prozesskomponenten, Ressourcen und des Layouts. Etabliert hat sich dafür das Bibliothekskonzept, das den Fertigungsplanern Standardmodule für die Entitätstypen Prozess und Ressource zur Verfügung stellt (vgl. ebenda S.16). Der Nutzen liegt in der Wiederverwendung von Bausteinen und bester Vorgehensweisen sowie der vereinfachten und einheitlichen Dokumentation der Planungsstände [VDI 4499, S.13]. Weitere Zeiteinsparungen sollen im Rahmen der Automatisierung in der Planung erzielt werden. Rooks nennt als Beispiele hierfür [Ro09, S.16]: Berechnungsalgorithmen für Logistikkosten, Kapazitätsabschätzungen bei Ladungsträgern, Parametrisierung von Lagerorten und die Erstellung von Ablaufsimulationen.

Die Standardisierung, Integration und Automatisierung werden im nachfolgenden Teil dieser Arbeit eine zentrale Rolle spielen. Betrachtet werden soll dabei die Integration von automatisierten Simulationsmodellgeneratoren in die Landschaft der betrieblichen Anwendungssysteme.

3 Automatisierte Simulationsmodellgenerierung

Die automatisierte Erstellung von Simulationsmodellen hat in den letzten zehn Jahren deutlich an Aufmerksamkeit gewonnen, was sich insbesondere in der Anzahl der Publikationen der vergangenen Jahre widerspiegelt. Ein Grund der diese Entwicklung begünstigte, ist die zunehmende Verbreitung der Digitalen Fabrik. Sie bietet mit ihrem einheitlichen und durchgängigen Datenmanagement sowie der Standardisierung von Planungsmodellen, -methoden und des Planungsablaufs gute Voraussetzungen für eine automatisierte Modellgenerierung.

Reinhart und Feldmann kommen im Rahmen einer Studie zum Einsatz der Simulationstechnik in Deutschland zu dem Ergebnis, dass der zeitliche und kapazitive Aufwand der Bereiche Datenbeschaffung und -aufbereitung mehr als die Hälfte des Gesamtaufwandes einer manuell durchgeführten Simulationsstudie ausmachen [ReFe97a, S.22]. Hrdliczka und Acél beziffern den zeitlichen Aufwand selbst für kleinere Simulationsprojekte auf 15-20 Arbeitstage [HrAc98, S.10]. Dies verhindert bei Projekten mit noch häufig zu erwartenden Änderungen den konsequenten Einsatz der Simulation. Darüber hinaus werden Simulationsmodelle bezüglich ihrer Modellierung oft an der konkreten Problemstellung ausgerichtet. Variationen des Modells müssen deshalb häufig mit hohem Aufwand neu modelliert werden [Ec02, S.206]. Demnach erscheint es naheliegend, den Prozess der Modellerstellung zu automatisieren. Eckardt unterstreicht die Notwendigkeit zur Automatisierung bei der Erstellung des Prozessabbildes von simulationsbasierten Prozessregelungsverfahren im Bereich der Produktionsplanung und -steuerung (PPS) [Ec02, S.2]. Rooks nennt die maschinelle Unterstützung der Modellerstellung für die zeitnahe dynamische Absicherung ohne zusätzliche Kapazitäten „unumgänglich“ [Ro09, S.55].

3.1 Voraussetzungen und Ziele

Bei der automatisierten Modellerstellung steht im Allgemeinen die Mehrfachverwendung des Generators im Vordergrund, da der wesentlich höhere Aufwand im Vorfeld, gegenüber einer einzelnen individuellen Lösung, erst hierdurch kompensiert werden kann. So muss im automatisierten Fall eine allgemeine Lösung des Problemfeldes gefunden werden, die eine ganze Reihe von Spezialfällen beinhaltet [Ro09, S.58]. Weiterhin sind Schnittstellenanbindungen zu betrieblichen Anwendungssystemen zur Eingangsdatenversorgung und ggf. Datenweitergabe zu realisieren. Auch die maschinelle Unterstützung bei der Aufbereitung und Zusammenführung der Eingangsdaten und Konsistenzprüfungen erhöhen den Aufwand. Eine automatisierte Modellerstellung bewirkt, aus zuvor genannten Gründen, zwangsläufig ein Mindestmaß an Standardisierung der erzeugten Modelle. Sie ist deswegen nur bei Fertigungssystemen anwendbar, die sich in Form standardisierter Modelle abbilden lassen. Hieraus ergibt sich eine Eingrenzung der abbildbaren Systeme und die Beschränkung auf

typische Fragestellungen [BFW00] zitiert nach [Wu00, S.113]. Weiterhin ist es wichtig, dass die Anwender, respektive die Fertigungsplaner, über ein Grundverständnis der Möglichkeiten und Grenzen der Simulationstechnik verfügen (vgl. ebenda S. 113).

Das Ziel der automatisierten Modellgenerierung besteht darin, den Simulationsexperten von Routineaufgaben, wie dem Platzieren und Parametrisieren von Objektentitäten, zu entlasten. Die hierdurch gewonnene Zeit kann dazu eingesetzt werden, weitere Planungsszenarien oder Simulationsprojekte zu realisieren [Ro09, S.57]. Vereinzelt wird auch der Wunsch geäußert, den Simulationsexperten teilweise ersetzen zu können [Ec02, S.3]. Im Bereich des operativen Einsatzes zielt die Automatisierung darauf, Änderungen des realen Prozessgeschehens zeitnah im Simulationsmodell nachzuvollziehen und das Modell so zu initialisieren, dass es dem aktuellen Zustand des realen Fertigungssystems entspricht [Ec02, S.2; Je07, S.129]. Eine Beschleunigung der Modellgenerierung wird vor allem in den Bereichen der Datenbeschaffung, -analyse bzw. -aufbereitung und der Modellierung erreicht [Ro09, S.55f; Je07, S.128]. Eine Unterstützung der Modellverifikation und -validierung ist ebenfalls zu erwarten, da basierend auf einem validierten Erstellungsprozess unter Verwendung von ebenfalls validierten Eingangsdaten das erzeugte Modell gleichfalls valide sein wird, solange die zu untersuchende Problemstellung einer Klasse zuzuordnen ist, die vom Modellgenerator abgedeckt wird. Rooks fasst folgende zu erwartende Vorteile einer automatisierten Modellgenerierung zusammen [Ro09, S.56f]:

- Verkürzter Simulationsmodellaufbau
- Schnellere Bewertung von Planungsalternativen
- Verbesserte Absicherung der Planungsergebnisse
- Zugriff auf valide und aktuelle Datenbasis
- Überprüfung der Planungsdaten auf Konsistenz
- Vereinfachung der Verifikation und Validierung durch Standardisierung
- Know-how-Transfer durch Speicherung von Expertenwissen in Bibliotheken

3.2 Anwendungsfelder von automatisierten Modellgeneratoren

Automatisierte Modellgeneratoren wurden im Wesentlichen für das Einsatzgebiet der Fertigungsplanung und zur Unterstützung im operativen Bereich während des Betriebes (PPS) entwickelt. Demnach lassen sich die Anwendungsfelder prinzipiell in planungs- und betriebsbegleitend unterscheiden. Der Schwerpunkt beim planungsbegleitenden Einsatz liegt in der Planungsabsicherung von Produktionseinrichtungen. Bei der betriebsbegleitenden Anwendung werden hingegen die Aufgaben der Produktionsplanung und -steuerung (PPS) unterstützt (vgl. Kapitel 2.2). Im Folgenden sollen diese beiden Anwendungsfelder näher beleuchtet und auf einige Ansätze aus der Literatur eingegangen werden.

3.2.1 Planungsbegleitender Simulationseinsatz

Der Einsatz planungsbegleitender Modellgeneratoren dient dazu, die Planungsphase von Produktionseinrichtungen zu unterstützen. Folgende Ziele werden hierbei verfolgt:

- Reduzierung des Aufwandes zur Erstellung der Modellvarianten
- Beschleunigung der Planung
- Vereinfachung des Simulationseinsatzes, insbesondere auch Verbesserung der Anwendbarkeit in frühen Phasen der Planung
- Verbesserte Planungsqualität mittels durchgängigen Simulationseinsatzes über die Planungsphasen

Ein Rahmen, in dem eine automatisierte Erstellung von Simulationsmodellen zunehmend an Interesse gewinnt, ist die Anwendung in der Digitalen Fabrik. Hier kommen zur Prozess- und Logistikplanung des Fertigungssystems rechnergestützte Planungstools zum Einsatz. Diese dienen der ganzheitlichen Fertigungsplanung und unterstützen den Prozess der parallelen Entwicklung von Produkt und Produktionseinrichtungen (Simultaneous Engineering) [Kü06, S.213]. Mit ihrer Hilfe kann das Fertigungskonzept, welches beschreibt, was, wie und mithilfe welcher Ressourcen gefertigt werden soll, digital erstellt werden. Das Fertigungskonzept führt die Elemente Produkt, dessen Verbindungsdaten der Einzelteile, Prozessbeschreibung und Ressource zusammen.

Da Simulationsanwendungen der Planung auf den Daten des Fertigungskonzeptes beruhen und Änderungen am Produkt oder Herstellungsprozess Auswirkungen auf das Simulationsmodell haben, können eine Vielzahl von Modellvarianten entstehen. Aus der Motivation heraus den Erstellungsprozess der Varianten zu beschleunigen, leitet Franke die Notwendigkeit ab, Ablaufsimulationen direkt aus dem Fertigungskonzept ableiten zu können [Fr03, S.87]. Eine Möglichkeit dies zu gewährleisten ist die automatisierte Modellgenerierung. In späteren Planungsphasen, mit zunehmender Konkretisierung der Datenlage und Reife des Fertigungskonzeptes, sehen Bley, Franke und Zenner die automatisierte Übertragung des Fertigungskonzeptes in ein Simulationsmodell als Ausgangsbasis für die Arbeit eines Simulationsexperten an, der das Grundgerüst zu einem individualisierten Modell ausbauen muss [BFZ03, S.321]. In früheren Phasen mit ausreichender Datengrundlage schlägt Franke die Verwendung von Grobsimulationen auf Basis standardisierter Modellbausteine vor. Diese sollten allerdings auf das Unternehmen und seine Fertigungsanlagen, sowohl auf Seiten des Planungs- als auch auf Seiten des Simulationstools, speziell angepasst sein [Fr03, S.91].

Bei der automatisierten Modellgenerierung basierend auf dem Fertigungskonzept rechnergestützter Planungstools, kann ein Großteil der benötigten Eingangsdaten aus den Planungstools übernommen werden. Daten, die dort nicht gepflegt werden, müssen jedoch zusätzlich aus anderen Anwendungssystemen, z. B. Datenbanken, bereitgestellt oder in einer Zwi-

schenschicht der Modellgenerierung manuell eingegeben werden. Zu diesen Daten können u. a. Schichtmodelle für Ressourcen, Strategien der Produktionssteuerung (vgl. Kapitel 2.3.5) oder die Experimentdaten zählen. Strategien werden bei den meisten Modellgeneratoren innerhalb der Simulationsanwendung durch eigens programmierte Bausteine oder vom Hersteller bereitgestellte Algorithmen vorgehalten, ggf. ist eine Parametrisierung dieser möglich. Nötig wird dies deswegen, weil Strategien und Abläufe in betrieblichen Planungssystemen in der Regel nicht oder nicht in einer für die Weiterverwendung geeigneten Form vorliegen [Se05, S.33] und somit keine Übertragung der algorithmischen Beschreibung stattfinden kann.

Aufgrund des Vorhaltens kann die Einbeziehung von vordefinierten Strategien in das Modell durch bloße Namens- sowie ggf. Parameterangabe erreicht werden.

Beispiele planungsbegleitender automatisierter Modellgeneratoren basierend auf Fertigungskonzepten von rechnergestützten Planungstools finden sich in [Ze06], [Fr07] und [Ro09].

3.2.2 Betriebsbegleitender Simulationseinsatz

Ein weiteres wichtiges Einsatzfeld der Ablaufsimulation ist die betriebsbegleitende Simulation (vgl. Kapitel 2.3.2). Hier liegt der Schwerpunkt hauptsächlich auf der Unterstützung der Produktionsplanung und -steuerung (PPS) [Se05, S.16]. Die Anwendung automatisierter Modellgeneratoren im betriebsbegleitenden Einsatzgebiet ist darin begründet, dass Änderungen des realen Prozessgeschehens oft umfangreiche Änderungen des Prozessabbildes (Simulationsmodells) nach sich ziehen [Fr02, S.2]. Der konsequente Simulationseinsatz erfordert jedoch das Simulationsmodell schnell und effizient an die aktuelle Betriebssituation der Fertigung anpassen zu können [Se05, S.21f]. Mithilfe automatisierter Modellgeneratoren soll der Aufwand zur Modellerstellung so weit reduziert werden, dass die betriebsbegleitende Simulation in die alltäglichen Abläufe der PPS eingebunden werden kann (vgl. ebenda S. 23).

Voraussetzung hierfür ist die systematische datentechnische Integration und organisatorische Einbindung der Simulation in die betrieblichen Abläufe [Kü06, S.89f; VDI 3633-5, S.5]. Damit verbunden ist die Einrichtung von Schnittstellen zu anderen betrieblichen Informationssystemen, da das Simulationsmodell mit den Angaben des aktuellen Ist-Zustandes des realen Fertigungssystems synchronisiert werden muss [VDI 3633-5, S.4]. Im Vordergrund stehen hierbei Schnittstellen zu folgenden Systemen: Enterprise Resource Planning (ERP) bzw. Produktionsplanung und -steuerung (PPS), Betriebsdatenerfassung (BDE) und Manufacturing Execution System (MES). Ein Überblick zu Anwendungen der Simulation im Bereich der PPS ist in Kapitel 2.3.2 zu finden.

Als Beispiele für betriebsbegleitende automatisierte Modellgeneratoren können die Ansätze von Splanemann [Sp95], Selke [Se05] und Jensen [Je07] angesehen werden.

3.3 Stand der Forschung zu automatisierten Modellgeneratoren

Dieser Abschnitt dient dazu, zwei der zuvor genannten Ansätze automatisierter Modellgeneratoren herauszugreifen und kurz vorzustellen. Hierbei sollen insbesondere Aspekte herausgestellt werden, welche für die beiden Ansätze als wesentlich angesehen werden können und auch im weiteren Verlauf der vorliegenden Arbeit von größerer Bedeutung sind. Dies betrifft im Besonderen die Anbindung der Simulation an das Umfeld betrieblicher Anwendungssysteme, die Art und Weise der Beschreibung des Simulationsmodells sowie die Integration von Strategien und Abläufen der Produktionssteuerung.

Betriebsbegleitender Ansatz von Selke [Se05]:

Selke setzt sich das Ziel, die Ablaufsimulation betriebsbegleitend für kurzfristige Zeithorizonte als ergänzendes Prognose- und Planungsinstrument in der Produktionsplanung und -steuerung einzusetzen. Seine Intension ist es, die Reaktionsfähigkeit von Produktionsunternehmen, welche schnell wechselnden Umfeldbedingungen ausgesetzt sind, zu erhöhen.

Aus dem Grund heraus, das Simulationsmodell schnell und effizient an die aktuelle Betriebssituation der Produktion anpassen zu müssen, setzt Selke zur Modellerstellung einen automatisierten Ansatz ein. Um einen möglichst hohen Realitätsbezug des Modells zu gewährleisten, legt er besonderen Wert darauf, die in der Produktion real ausgeführten Strategien und Abläufe im Simulationsmodell abzubilden (vgl. hierzu Kapitel 2.3.5). Da diese in betrieblichen Planungssystemen in der Praxis nicht explizit und in einer nicht weiterverwendbaren Form vorliegen [Se05, S.33], nutzt Selke zur Identifikation der Strategien und Abläufe das Datenmaterial aus Betriebsdatenerfassungssystemen⁶ (BDE-Systeme).

Untermauert wird dieser Ansatz dadurch, dass in der Praxis Situationen auftreten können, bei denen weder Art noch Ausführung der angewendeten Strategien vorhersehbar sind, z. B. Störungen im Betriebsablauf auf die mit individuellen Notfallstrategien reagiert werden muss [Se05, S.49]. Diese können daher zwangsläufig nicht aus betrieblichen Planungssystemen ausgelesen werden. Aus diesem Grund setzt Selke Methoden der Mustererkennung ein, um zu einer Interpretation der real angewendeten Strategien und Abläufe zu gelangen. Diese werden in einem eigens entwickelten Metamodell kodifiziert. Durch die Beschreibung in jenem Metamodell verwaltet er die erkannten Strategien und macht sie für die automatisierte Modellgenerierung nutzbar.

6 Unter einem System der Betriebsdatenerfassung wird „eine Menge von Hard- und Softwarekomponenten verstanden, die Erfassungs- und Ausgabefunktionalität für betriebliche Daten mit Hilfe von automatisch arbeitenden Datengebern (Sensoren) oder manuell bedienbaren Erfassungsstationen in einem Produktionsbetrieb zur Verfügung stellen“ [Mö08]. Die erfassten betrieblichen Daten lassen sich den folgenden Kategorien zuordnen [Ku05, S.299f]: Auftragsdaten, Personaldaten, Betriebsmitteldaten, Werkzeug- und Vorrichtungsdaten, Lager- und Materialdaten und Qualitätsdaten.

Um aus den BDE-Daten die Anwendung bestimmter Strategien ableiten zu können, wird auf eine Clusterung der Strategien zurückgegriffen.⁷ Für diese Cluster wird, unter Verwendung charakteristischer Eigenschaften (Mustermerkmale), ein passendes Muster erstellt.⁸ Im Rahmen der Mustererkennung werden diese Mustermerkmale dann dazu verwandt, die eingesetzten Strategien zu identifizieren und einer der drei Strategiecluster zuzuordnen. Darauf aufbauend erfolgt eine Spezifizierung innerhalb der Cluster mit dem Ziel, die bis jetzt nur entsprechend der drei Cluster zugeordnete Strategie qualitativ zu verfeinern und quantitativ zu beschreiben. Im Fall erfolgreichen Detektierens liegt im Ergebnis eine Beschreibung der Strategie mittels des Metamodells vor, die besagt, wie die Strategie ausgeprägt ist bzw. wann und unter welchen Bedingungen sie eingesetzt wird.

Eine Zusammenfassung des Vorgehens von Selke zur Interpretation von Strategien und Abläufen der Produktionssteuerung ist in Abbildung 12 dargestellt.

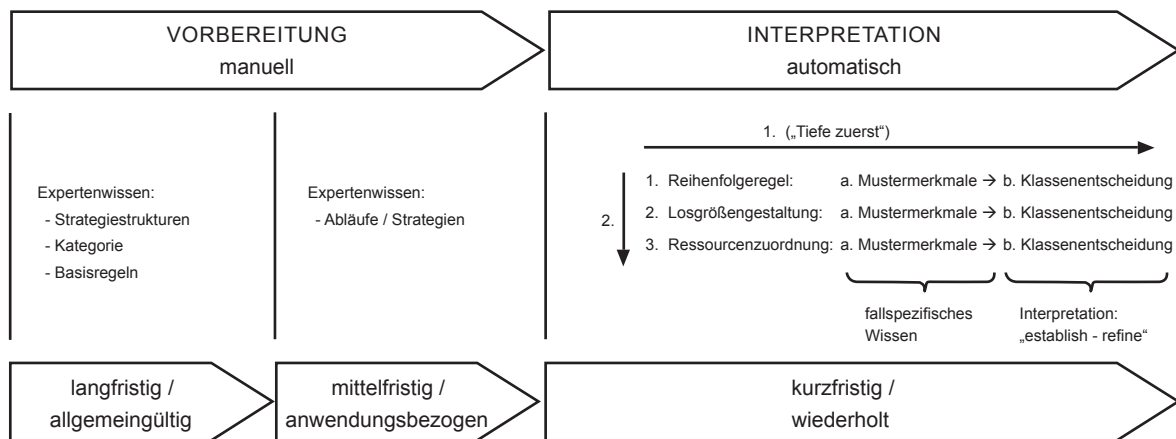


Abbildung 12: Vorgehensmodell zur Interpretation von Strategien und Abläufen der Produktionssteuerung [Se05, S.95]

Der eigentlichen Interpretation ist eine Vorbereitungsphase vorangestellt. In dieser wird die Grundstruktur der Strategiecluster erstellt und die dazugehörigen charakteristischen Eigenschaften definiert. Um der Überlagerung und Kombinierbarkeit der Basisregeln⁹ Rechnung zu tragen, kann die Grundstruktur mehrstufig aufgebaut werden. Der anwendungsbezogene Teil der Grundstruktur ist dabei regelmäßig den betrieblichen Erfordernissen anzupassen. Die Eingangsdaten werden aus BDE- oder PPS-Systemen extrahiert und in Dateiform gespeichert. Anschließend erfolgt die Auswahl simulationsrelevanter Daten in Microsoft Excel unter Verwendung der Makrosprache Visual Basic und die Interpretation der Strategien und Abläufe. Nach der Interpretation wird dem Nutzer ein aufbereitetes Ergebnis dialogbasiert angezeigt, wobei noch Änderungen vorgenommen werden können. Alle simulationsrelevan-

7 Nach der Einteilung gemäß Tabelle 1 aus Kapitel 2.3.5, lassen sich Strategien und Abläufe der Strategiecluster Losgrößengestaltung, Ressourcenbelegung, Reihenfolgegestaltung und Auftragsfreigabe zuordnen.

8 Auf die Betrachtung der Strategien zur Auftragsfreigabe verzichtet Selke in seiner Arbeit.

9 Als Basisregel wird eine Regel bezeichnet, die nicht weiter unterteilt werden kann. Im Strategiecluster der Reihenfolgegestaltung sind z. B. FIFO und KOZ Basisregeln.

ten Daten werden in einer Microsoft Access Datenbank gespeichert und mittels ODBC dem Simulator Plant Simulation übergeben. Der eigentliche Modellaufbau erfolgt automatisiert im Simulator.

Für den alltäglichen Einsatz ist der realisierte Prototyp, nach eigenen Aussagen, noch nicht anwendbar, da insbesondere ein intuitiv bedienbarer Modell- und Versuchseditor nicht implementiert wurde [Se05, S.120]. Einen weiteren Schwachpunkt stellt die dateibasierte Eingangsdatenversorgung dar.

Planungs-/ betriebsbegleitender Ansatz von Jensen [Je07]:

Jensen stellt einen automatisierten Ansatz zum planungsbegleitenden Einsatz der Simulation vor, der in späteren Ausbaustufen auch betriebsbegleitend zur Unterstützung der Produktionsplanung und -steuerung eingesetzt werden soll. Sein Ansatz ist aus der Motivation heraus entstanden, den Erstellungsaufwand für den Simulationseinsatz so weit zu reduzieren, dass die Simulation in den Planungsalltag integriert werden kann. Hierfür soll weiterhin erreicht werden, dass die Pflege der Simulationsmodelle vereinfacht wird und sie somit ihre Aktualität in Bezug zum realen Fertigungssystem behalten und über den Lebenszyklus der Anlage hinweg eingesetzt werden können. In Abbildung 13 ist sein prinzipielles Vorgehen skizziert.

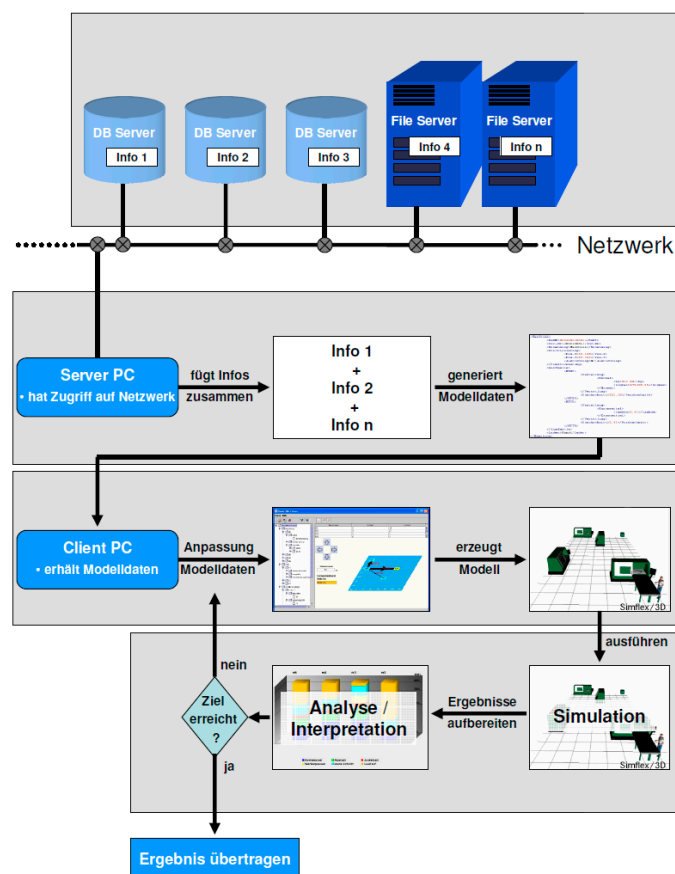


Abbildung 13: Vorgehensmodell des Ansatzes von Jensen [Je07, S.5]

Eine Analyse des bestehenden Datenmaterials während seiner praktischen Tätigkeit an einem Unternehmensstandort ergab, dass ein Großteil, der für Simulationsstudien benötigten Daten bereits vorhanden ist und in verschiedenen Datenbanksystemen verwaltet wird. Deshalb konzentrierte sich Jensen darauf, diese Daten aus den unterschiedlichen Systemen auszulesen, zu bündeln und in einem Datenmodell strukturiert abzulegen.

Hierfür wurde ein Datenframework entworfen, welches auf Basis von Web-Diensten über Anbindungsmodule zu den einzelnen Datenquellen verbunden ist und die benötigten Informationen flexibel zusammenstellen kann. Dabei möchte Jensen auch anderen Anwendungssystemen außer den Simulationssystemen ermöglichen, auf die verteilten Daten über das Framework zugreifen zu können.

Für die Datenzusammenführung, -darstellung und -übermittlung hat sich Jensen für das XML-Datenformat entschieden und eine Struktur zur Modellbeschreibung erarbeitet, welche, da auf XML basierend, leicht erweiterbar ist.

Der Server übernimmt die Aufgabe, die Anfragen der Client-Rechner entgegenzunehmen und daraufhin Verbindung zu den Datenquellen aufzunehmen (vgl. Abbildung 13). Anschließend werden die Daten durch den Server zusammengetragen, aufbereitet und den Clients zur Verfügung gestellt. Auf den Client-Rechnern bestehen die Möglichkeiten zur grafischen Darstellung und Bearbeitung der XML-Datenstrukturen, bevor sie zu den Simulationssystemen übertragen werden. Realisiert wurden Anbindungen zu den Systemen Quest, Simflex/3D und Automod. Innerhalb der Simulationssysteme findet der Modellaufbau automatisiert unter Verwendung vordefinierter Bausteinbibliotheken statt, in welchen auch die Steuerstrategien enthalten sind.

In der praktischen Anwendung konnte nicht auf die Unterstützung von Simulationsexperten verzichtet werden, da trotz Automatisierung viel Erfahrung nötig war, um leistungsfähige und valide Modelle zu erstellen sowie aus den erzielten Ergebnissen die richtigen Schlüsse zu ziehen. Auch zeigte sich, dass kein Modell wie das andere ist und nur gelegentlich standardisierte Grundmodelle für ähnliche Zielsetzungen geschaffen werden konnten. Der Modellierungsaufwand wurde nach eigenen Aussagen durch die Nutzung des Frameworks jedoch deutlich verringert [Je07, S.127f].

In seinen Schlussfolgerungen verleiht Jensen der Forderung nach einer standardisierten Modellbeschreibung für Simulationsmodelle Nachdruck [Je07, S.129]. Das ist insoweit von Bedeutung, da im Rahmen der vorliegenden Arbeit untersucht und im praktischen Teil auch ausprobiert werden soll, ob mit dem Core Manufacturing Simulation Data (CMSD) Information Model diese Forderung erfüllt werden kann.

3.4 Fazit

In beiden vorgestellten Ansätzen zur automatisierten Simulationsmodellgenerierung gelingt, wie in vielen anderen Beiträgen auch, keine vollautomatische Generierung soweit die Grenzen der Funktionsabdeckung nicht sehr eng gezogen werden. Das liegt u. a. darin begründet, dass bestimmte Datenumfänge, welche zu einer hohen Detailgenauigkeit des Modellabbildes beitragen, nicht in rechnergestützten Anwendungssystemen gepflegt werden. Dazu zählen insbesondere Strategien und Abläufe, die sich in subjektiven Entscheidungsbereichen von Meistern und Werkern befinden. Diese müssen erst durch Befragungen oder Beobachtungen aufgenommen werden [Je07, S.107]. Beispiele hierfür sind u. a. Notfallstrategien bei Störungen oder Priorisierungsstrategien bei der Maschinenauswahl.

Eine allgemeingültige, allumfassende Lösung zur Automatisierung der Modellbildung konnte bisher weder gefunden noch konzeptionell vollständig entwickelt werden [Ec02, S.50]. Einen Hauptgrund dafür sieht Eckardt in der Schwierigkeit Verfahrensansätze zu finden, die in der Lage sind, die Komplexität operativer Produktionsprozesse vollständig zu erfassen, hinreichend abzubilden und automatisiert in Simulationsmodelle zu überführen (vgl. ebenda S. 50). Eine vollständige Automatisierung der Modellbildung auf dem Niveau eines erfahrenen menschlichen Modellierers wird somit bis auf Weiteres nicht möglich sein (vgl. ebenda S. 212). Diese These soll noch durch das folgende Zitat untermauert werden.

„Große Freiheitsgrade hinsichtlich der Modellierungsgenauigkeit und der gewünschten Zielstellung eines Simulationsmodells sowie der Auswahl der heterogenen Daten- und Informationsquellen erfordern in hohem Maße die Fähigkeit des Modellierers, das Problem auf abstraktem Niveau zu Durchdringen, zu Verstehen und darauf aufbauend zu modellieren“ (vgl. ebenda S. 211f).

Weiterhin geht mit der Automatisierung der Modellerstellung zwangsläufig eine Standardisierung und infolgedessen ein Einbüßen von Flexibilität einher. Da aber das Spektrum der zu untersuchenden Sachverhalte mit deren Zielstellungen sehr vielschichtig ist und vorab nicht allumfassend eingegrenzt werden kann, muss eine Beschränkung der durch die automatisierte Erstellung abgedeckten Aufgabenstellungen erfolgen. Eine universelle Methode, die jegliche Problemstellungen berücksichtigt, kann demnach nicht gefunden werden.

Dennoch zeigen die zahlreichen Ansätze, dass automatisierte Modellgeneratoren innerhalb festgelegter Grenzen möglich sind und einen positiven Beitrag zur Verbreitung der Simulationstechnik liefern können. Weiterhin lässt sich die Erkenntnis gewinnen, dass leistungsfähige Schnittstellenkonzepte nötig sind, um der Forderung nach einer umfassenden und flexiblen Einbindung in die Landschaft der betrieblichen Anwendungssysteme nachkommen zu können. Dem XML-Datenformat wird in diesem Zusammenhang oft ein gewichtiger Beitrag zu dieser Erfüllung vorhergesagt.

4 Integration der Simulation im betrieblichen Umfeld durch das Core Manufacturing Simulation Data (CMSD) Information Model

Die Forderung nach leistungsstarken Schnittstellen zwischen betrieblichen Anwendungs- und Simulationssystemen wird zusehends lauter. Zur Lösung des Problems werden gerade in jüngster Vergangenheit Modellbeschreibungen basierend auf dem XML-Format favorisiert. Einige Autoren lieferten in ihren Beiträgen bereits Vorschläge für Datenmodelle mit der Absicht, dass diese zur Diskussion anregen und zukünftig zu Standards ausgebaut werden könnten (vgl. hierzu z. B. Ro09, Je07, Sp95). In diesem Kapitel soll auf eine weitere Entwicklung diesbezüglich eingegangen werden.

Im Jahr 2006 wurde eine erste Vorabversion des Core Manufacturing Simulation Data (CMSD) Information Model der Öffentlichkeit vorgestellt. Seitdem erfuhr die Spezifikation umfangreiche Änderungen und steht nun in einer Version vom April 2010 zur Verfügung, die als Standard eingebracht werden soll. Im Folgenden wird die Motivation und Zielsetzung dieser Initiative herausgearbeitet, um anschließend den Aufbau näher zu betrachten. Hierbei soll auf die wesentlichen Pakete der Spezifikation eingegangen werden, die auch insbesondere für die prototypische Umsetzung eines automatisierten Modellgenerators im darauffolgenden fünften Kapitel verwendet wurden. Abschließend werden die Potenziale des Standards herausgearbeitet und es wird auf potenzielle Schwachstellen hingewiesen.

4.1 Motivation und Zielsetzung

Die Simulationstechnik ist unbestritten eine etablierte Technologie, welche es ermöglicht, Kosten bei der Planung oder während des Betriebs von Produktionsanlagen zu senken, die Qualität der Produkte und der Planung zu verbessern sowie die Produkteinführungszeit (time to market) zu verkürzen. Dennoch existieren ökonomische Barrieren, die den konsequenten Einsatz der Simulation in der Industrie behindern. So ist der Einsatz der Simulation mit nicht unerheblichen zeitlichen und finanziellen Aufwänden verbunden. Einen wesentlichen Aspekt hierbei stellt die Integration der Simulation in das Umfeld der betrieblichen Anwendungssysteme dar [Jo07, S.1673f]. Dies hat dazu geführt, dass immer wieder Anstrengungen unternommen wurden, die Schnittstellenanbindung der Simulation zu verbessern.

Eine hiervon ist die Bildung der Core Manufacturing Simulation Data (CMSD) Product Development Group, welche unter den Leitlinien der Simulation Interoperability Standards Organization (SISO) und in Zusammenarbeit mit industriellen und universitären Partnern eine Datenspezifikation erstellt hat, die einen effizienten Datenaustausch zwischen betrieblichen Anwendungssystemen und Simulationsanwendungen ermöglichen soll [LLR06, S.1]. In der Spezifikation ist ein Datenmodell mit Grundstrukturen definiert, mit deren Hilfe Modellbeschreibungen für Produktionsdaten formuliert werden können. Der Fokus liegt hierbei

darauf, die Simulation ins Umfeld der Anwendungssysteme zu integrieren. Dementsprechend wurden die Datenstrukturen danach ausgerichtet, simulationsrelevante Daten abbilden zu können.¹⁰ Mittels dieser Beschreibung ist es möglich, den Datenaustausch zwischen Anwendungssystemen zu gewährleisten. Das primäre Ziel ist es, diese Datenstrukturen über den gesamten Produktionsprozess nutzen zu können [SISO10, S.10]. Es soll also möglich werden, ein und dieselbe Datenbeschreibung für die Produktionsplanung und -steuerung und darüber hinaus für die Datenversorgung der angeschlossenen Simulationssysteme zu verwenden (vgl. Abbildung 14). Hierdurch sollen notwendige Datentransformationen bei der Anbindung von Anwendungs- und Simulationssystemen weitestgehend minimiert werden [LLR06, S.3]. Als Datenformat zur Übertragung wurde das XML-Format ausgewählt. Dieses zeichnet sich durch Systemunabhängigkeit und der Möglichkeit aus, Daten in strukturierter Form ablegen zu können [Ro09, S.121].

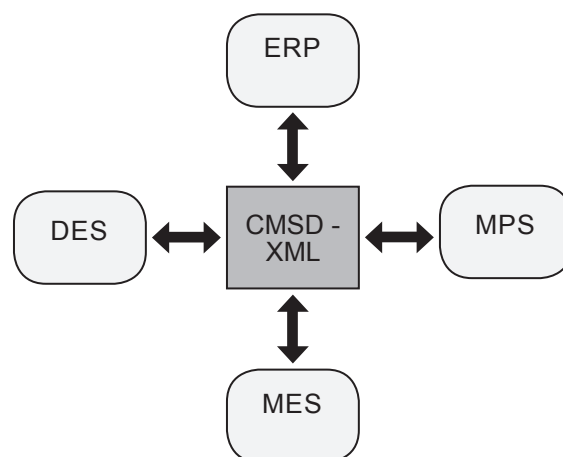


Abbildung 14: CMSD als neutrales Datenformat zwischen betrieblichen Anwendungssystemen [Jo07, S.1674]

Die Ziele des CMSD Information Model im Einzelnen sind [SISO10, S.10]:

- Fördern der Entwicklung und Nutzung von Simulationsanwendungen bei Fertigungsabläufen
- Vereinfachen des Datenaustausches zwischen betrieblichen Anwendungs- und Simulationssystemen
- Ermöglichen besserer Test- und Evaluierungsverfahren von Fertigungssoftwaresystemen
- Verbessern der Interoperabilität von Fertigungsanwendungssystemen

¹⁰ Eine Einteilung simulationsrelevanter Daten ist im Kapitel 2.3.5 zu finden.

4.2 Aufbau und Struktur

Die Spezifikation des Standards besteht aus zwei Teilen. Der erste Teil ist eine grafische Repräsentation der Pakete und Klassen, die unter Verwendung der Unified Modeling Language (UML) formuliert wurden. Die grafische Darstellungsweise in Form von Klassendiagrammen erleichtert es dem Anwender, die Verknüpfungen der Datenstrukturen untereinander zu verstehen und eignet sich daher sehr gut zur Dokumentation der Spezifikation. Darüber hinaus bieten leistungsstarke UML-Werkzeuge die Möglichkeit, automatisiert XML-Schemata aus Klassendiagrammen generieren zu können.

Im zweiten Teil erfolgt eine Beschreibung des Modells mithilfe von XML-Schemata, welche dazu genutzt werden, die eigentlichen XML-Dateien für den Datenaustausch zu erzeugen. Erweitert wurden die XML-Schemata um eingebettete Schematron¹¹ Schemata, damit alle Restriktionen aus der UML-Version berücksichtigt werden konnten [RiLe08, S.1778]. Diese Konstellation erlaubt es, automatisiert CMSD XML-Dateien zu generieren, die entsprechend des Standards stets valide sind. Im Folgenden wird zunächst der UML-Teil der Spezifikation im Fokus stehen, konkrete Ausführungen zu dem auf XML basierenden Teil finden sich im anschließenden fünften Kapitel.

Das CMSD-Modell ist als eine Sammlung von untereinander in Beziehung stehenden Klassen modelliert, welche entsprechend ihrer Zugehörigkeit in Pakete gegliedert sind (vgl. Abbildung 15). Die Hauptfunktion der Pakete besteht darin, eine Gruppierung von Informationen aus unterschiedlichen Bereichen der Produktion vorzunehmen und eine übersichtliche, sinnvolle Gliederung zu erreichen [SISO10, S.16].

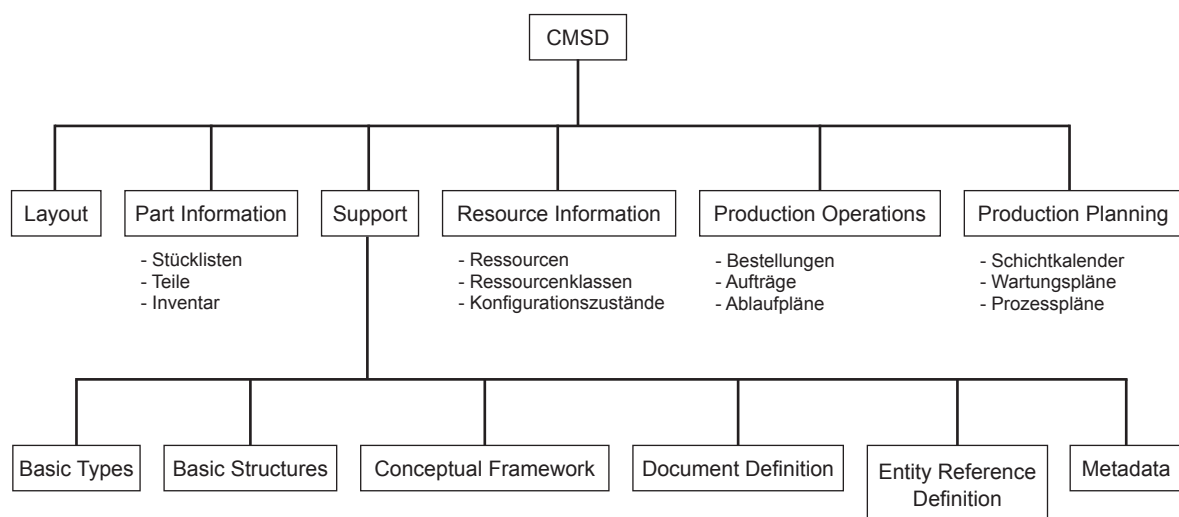


Abbildung 15: Pakete des CMSD-Modells nach [SISO10, S.18]

¹¹ Schematron ist eine regelbasierte Sprache, mit deren Hilfe die Einhaltung von Restriktionen in XML Dokumenten sichergestellt werden kann [RiLe08, S.1778f]. Sie dient damit der Validierung von Inhalt und Struktur.

Die Entitätstypen innerhalb der Pakete werden als Klassen und die Eigenschaften der Entitätstypen als Klassenattribute repräsentiert. Für die Entitätstypen wurde kein Verhalten definiert, d. h. sie werden lediglich durch ihre Attribute und Beziehungen zu anderen Entitätstypen charakterisiert. Restriktionen sind durch Vielfachheiten der Klassen und Attribute, Regeln, die textuell hervorgehoben sind, oder Bedingungen der Beziehungen abgebildet [SISO10, S.16f].

Innerhalb der nachfolgenden Abschnitte werden die einzelnen Pakete des CMSD Information Model kurz vorgestellt, wobei allerdings nur auf eine Auswahl der Klassen näher eingegangen werden soll.

4.2.1 CMSD Package

Das CMSD Paket steht auf der höchsten Ebene und enthält selbst keine eigenen Klassen. Es dient somit der Gruppierung aller im CMSD Information Model definierten Pakete, in denen wiederum die Klassen festgelegt sind. Auf der obersten Gliederungsebene stehend, schließt es alle anderen Pakete ein. Jedes dieser Pakete ist ausgerichtet auf einen zusammenhängenden Bereich von Klassen und Beziehungen eines spezifischen Teilbereiches des CMSD Information Model (vgl. ebenda S. 66).

4.2.2 Support Package

Basic Types Package

Das Support Paket ist zur übersichtlichen Strukturierung selbst noch einmal in sechs Unterpakete gegliedert (vgl. Abbildung 15). Diese Unterpakete enthalten u. a. Definitionen für Grundtypen und Basisstrukturen, die in anderen Paketen verwendet werden. Dies sorgt dafür, dass redundante Formulierungen für wiederkehrende Strukturen vermieden werden.

Beispiele für Grundtypen sind u. a. Aufzählungstypen für die Einteilung der Ressourcen, Längen- und Zeiteinheiten (vgl. Abbildung 16).

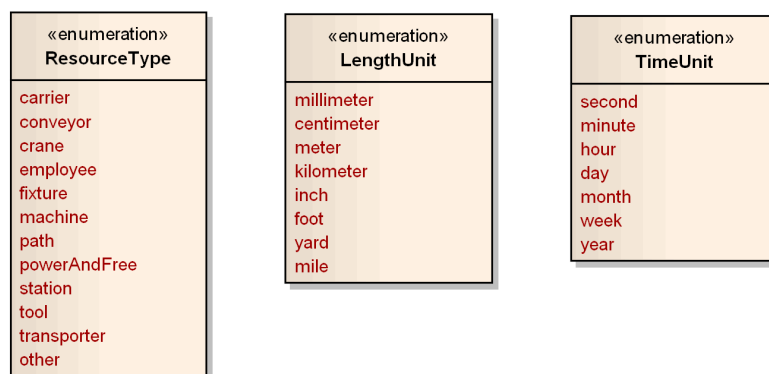


Abbildung 16: Beispiele für Grundtypen des CMSD Information Model [SISO10, S.54 u.57]

Entity Reference Definition Package

Dieses Paket übernimmt eine ganz wichtige Funktion. Und zwar definiert es Klassen, zu denen Verweise (Referenzen) erstellt werden können. Das ist immer dann wichtig, wenn innerhalb von Entitäten auf andere Entitäten verwiesen werden muss, um komplexe Sachverhalte abbilden zu können und die Verbindung noch nicht vordefiniert zur Verfügung steht. Genutzt werden kann dies z. B. dazu, um innerhalb einer Entität Maschine einen Verweis auf ihren Standort in der Fabrik zu definieren. Die Möglichkeiten Referenzen zu setzen sind dabei sehr vielfältig und bieten viel Spielraum für Kreativität.

Innerhalb des Pakets werden zwei Gruppen von Entitätsreferenzen unterschieden. Einmal die „Basic“ Entitätsreferenzen, welche Verweise zu Elementen definieren, die nur ein „Identifier“ Attribut (eindeutigen Bezeichner) besitzen (vgl. Abbildung 17). Zum anderen die „Complex“ Entitätsreferenzen, deren Zielelemente durch mehrere „Identifier“ Attribute charakterisiert werden (vgl. Abbildung 18). Ein Beispiel ist der Prozessplan, der einerseits selbst durch einen eindeutigen Bezeichner identifiziert wird, aber andererseits Prozessschritte beinhaltet, die ebenfalls einen eindeutigen Bezeichner innerhalb der Entität benötigen.

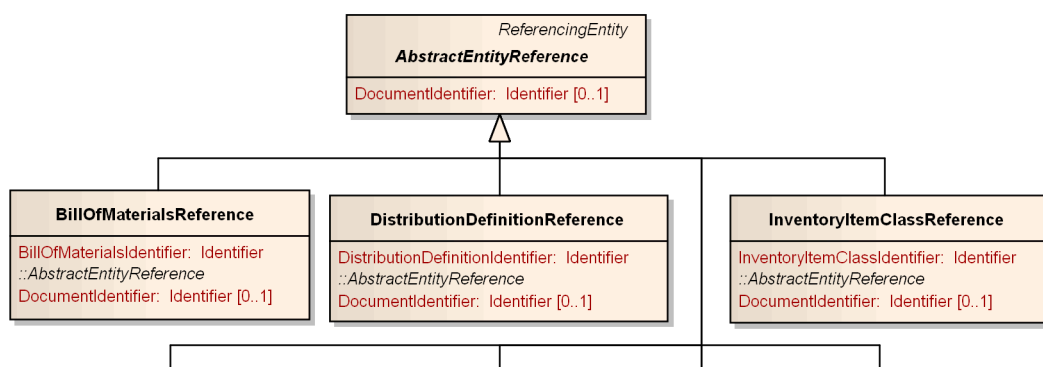


Abbildung 17: Ausschnitt der Basic Entitätsreferenzklassen [SISO10, S.79]

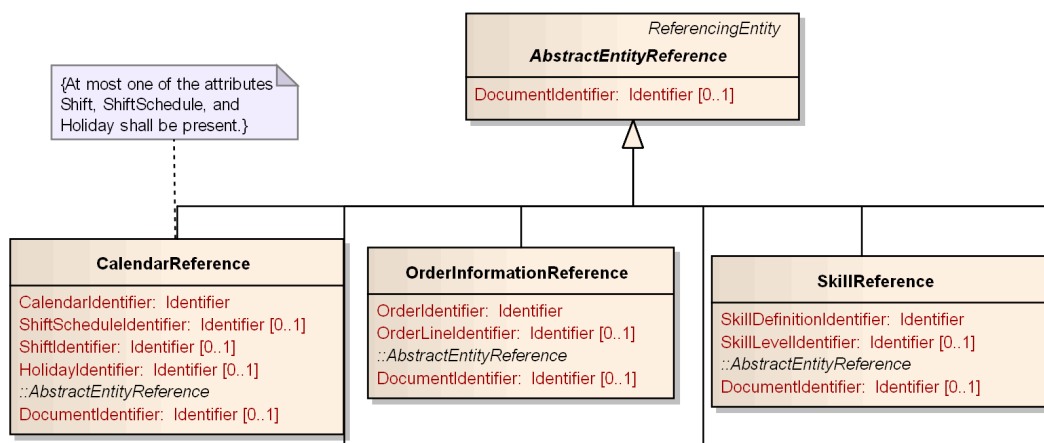


Abbildung 18: Ausschnitt der Complex Entitätsreferenzklassen [SISO10, S.81]

Basic Structure Package

Eine interessante Datenstruktur befindet sich im Paket der Basisstrukturen, das Konzept der Property. Auf sie soll nachfolgend eingegangen werden. Die Property ist dazu gedacht, Klasseneigenschaften zu definieren, welche durch die Klassenbeschreibung noch nicht vorgegeben sind. Das ist deswegen von besonderer Bedeutung, weil auf die Weise eine Anpassung der vorgegebenen Datenstrukturen an die eigenen Bedürfnisse möglich ist. Eine strikte Abhängigkeit von vordefinierten Eigenschaften bzw. Attributen wird somit umgangen. Da die Property in einer Vielzahl der komplexeren Klassen als Attribut zur Verfügung steht und dieses beliebig oft innerhalb der Klasse eingesetzt werden kann, besteht die Möglichkeit für diese Klassen beinahe uneingeschränkt viele Eigenschaften abzubilden. Die Abbildung 19 gewährt einen Einblick in die Definition der Property Klasse.

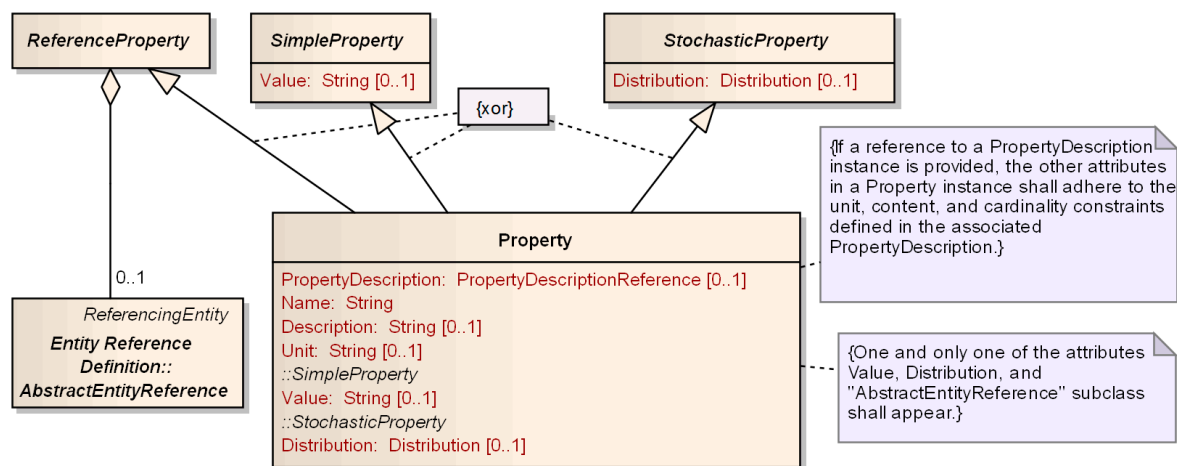


Abbildung 19: Property Klasse des Basic Structure Package [SISO10, S.21]

Die Property Klasse stellt Attribute für die Eingabe von Beschreibungen, Name, Einheit und Verteilung bereit. Bei der Nutzung einer Property ist auszuwählen, welche Art von Property verwendet werden soll. Folgende drei Möglichkeiten stehen zur Auswahl:

- einfache Property, bei der lediglich ein Wert des Typs String abgebildet wird
- stochastische Property, mit deren Hilfe stochastische Verteilungen angegeben werden können
- Referenz Property, die einen Verweis auf andere Entitäten ermöglicht

Mit Letzterem kann auf die Entitäten des Entity Reference Definition Package verwiesen werden. Vollständige Klassendiagramme dieses Pakets sind im Anhang A.1 zu finden.

4.2.3 Layout Package

Das Layout Paket ist erst im späteren Verlauf der Entwicklungsphase in die Spezifikation eingeflossen. Es wurde entwickelt, um zusätzlich zu den zahlreichen simulationsrelevanten Informationen, über z. B. Arbeitspläne, Aufträge und Ressourcen, auch grafische Beschreibungen von Objekten abbilden und transportieren zu können. Diese Informationen können u. a. in Simulationsprojekten zur Bewertung unterschiedlicher Layoutvarianten eine Rolle spielen [RiLe08, S.1]. Die zentralen Klassen des Layout Pakets sind in Abbildung 20 dargestellt.

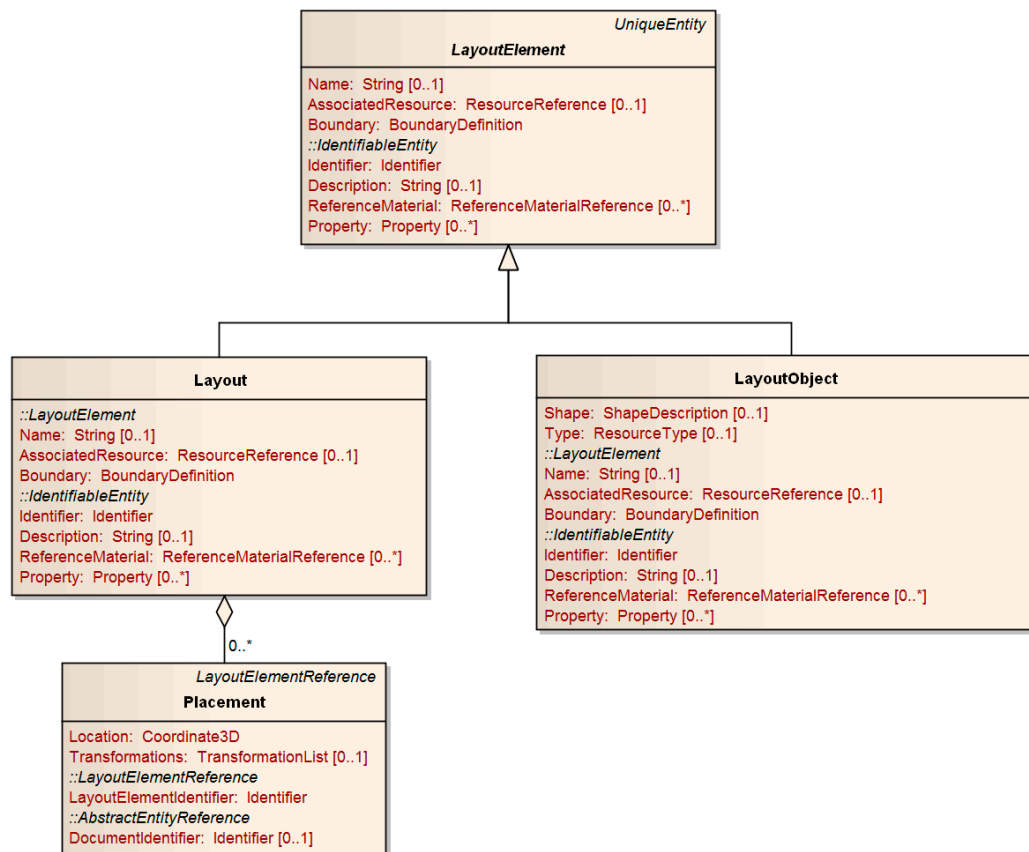


Abbildung 20: Klassendefinitionen der Layoutelemente [SISO10, S.88]

Umgesetzt wurde das Klassenkonzept mithilfe einer abstrakten Klasse „LayoutElement“ als Basisklasse, die grundlegende Eigenschaften für die von ihr abgeleiteten, spezialisierten Klassen bereitstellt.

Die Layout Klasse spezifiziert Informationen über einen Bereich in dem Produktionsaktivitäten stattfinden, beispielsweise die Halle in der die Fertigungsanlagen stehen. Dies beinhaltet Informationen zu den Abmaßen der Entität und der Orientierung des Koordinatensystems. Um nun Objekte innerhalb des Raumes eines Layouts abzubilden, werden Instanzen der Klasse „Placement“ (Platzierung) eingesetzt. Diese Platzierungen werden durch Angabe ihrer Koordinaten im Raum referenziert. Weiterhin ist es möglich, für sie Verschiebungen,

Drehungen und Skalierungen festzulegen. Durch einen Verweis auf eine Instanz der Klasse „LayoutObject“ erfolgt die Verbindung der Platzierungen mit ihrer Beschreibung.

Layoutobjekte repräsentieren Elemente, die ein Teil der Fertigungsanlage sein können. Neben den grundlegenden Angaben, wie Name und Ressourcentyp, ist es möglich, die Abmaße und die Orientierung des Koordinatensystems sowie elementare Formbeschreibungen festzulegen. Es steht eine Auswahl von vier möglichen Formbeschreibungen zur Verfügung (vgl. Abbildung 21):

- Angabe einer Basisform, wie Rechteck, Kreis und Polygon, im 2-dimensionalen Fall
- Informationen zu einem 2D-Bild oder 3D-Grafikmodell
- Unterteilung in mehrere Segmente, wobei auch krummlinige möglich sind
- Textuelle Bemerkung, die innerhalb des Layoutobjektes angezeigt werden soll

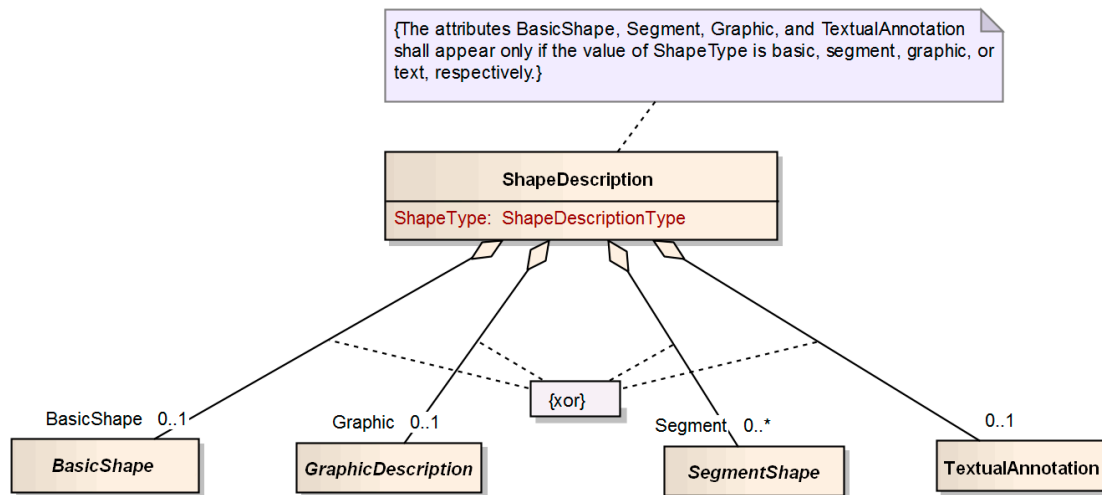


Abbildung 21: Klassendefinitionen der Formbeschreibungen [SISO10, S.89]

Zusammenfassend kann konstatiert werden, dass das Layout Paket Möglichkeiten zur groben räumlichen Beschreibung von Objekten bietet. Positiv hervorzuheben ist die Variante der Segmentierung von Elementen und die Berücksichtigung von gebogenen Formen, was bei der Modellierung von z. B. Transporteinrichtungen eine Rolle spielt. Darüber hinaus wurde durch die Möglichkeit von Transformationen, der im Layout befindlichen Elemente, bewerkstelligt, dass eine Abbildung eines groben Anlagenlayouts aus einzelnen Anlagenteilen möglich ist.

Was mit dem Layout Paket nicht beabsichtigt wurde, ist in Konkurrenz zu bestehenden CAD-Formaten bezüglich deren Abbildungsleistung zu treten. Somit ist es nicht für Anwendungen geeignet, bei denen sehr genaue Geometrien benötigt werden [ReLi08, S.1779f].

4.2.4 Production Operations Package

In diesem Paket werden Klassendefinitionen zur Beschreibung von externen Bestellaufträgen (Orders), betriebsinternen Aufträgen (Jobs) und Ablaufplänen (Schedules) bereitgestellt. Die vordefinierten Strukturen der Bestellauftragsabwicklung ermöglichen neben der Abbildung von typischen Kundenauftragsdaten, wie Preis, Anzahl, Auftragsstatus und Priorität, auch die Aufnahme von zusätzlichen Informationen über geordnete Services oder bestellte Waren (z. B. Verpackungsvorgaben) [SISO10, S.120f].

Zu den betriebsinternen Aufträgen gehören u. a. Bearbeitungs-, Wartungs- und Transportaufträge, die innerhalb des Fertigungssystems beauftragt werden. Hierbei sind z. B. Informationen darüber von Bedeutung, wer die Aufträge angefordert hat, wie Aufträge miteinander in Verbindung stehen und welche Informationen wichtig sind, um die Aufträge erfüllen zu können. Weiterhin ist zu unterscheiden, ob ein Auftrag bereits bearbeitet wird oder nur eingeplant ist. Wichtige vordefinierte Strukturen können Informationen zu Start- und Endzeitpunkt, Prozesszeit, benötigten Ressourcen und Teilen, Prozessplan und Wartungsplan sowie Kosteninformationen aufnehmen (vgl. ebenda S. 121). Die Klassendiagramme des Production Operations Package sind im Anhang A.2 zu finden.

4.2.5 Production Planning Package

Das „Production Planning Package“ beinhaltet Klassenstrukturen zur Modellierung von Schichtkalendern, Prozess-, und Wartungsplänen. Die Strukturen des Prozessplanes sollen nachfolgend kurz erläutert werden (vgl. hierzu Abbildung 22).

Der Prozessplan ist so gegliedert, dass innerhalb eines Planes einzelne Prozessschritte definiert werden, welche zusätzlich noch hierarchisch in Unterprozessgruppen zusammengefasst werden können. Im Prozessplan, wie auch in der Beschreibung der Prozessschritte, können Informationen zu den produzierten und benötigten Teilen, notwendigen Ressourcen und Kosteninformationen abgelegt werden.

Innerhalb eines Prozessschrittes stehen darüber hinaus u. a. folgende Strukturen zur Verfügung: Verweise zu Maschinenprogrammdateien, Attribute für Rüst- und Prozesszeit sowie Möglichkeiten, um Prozessrestriktionen festzulegen. Zur Festlegung von Restriktionen stehen nachfolgende Optionen zur Verfügung (vgl. hierzu die Klassendefinitionen im Anhang A.1):

- Festlegen des Vorgänger- und Nachfolgerprozessschrittes
- Festlegen der Anordnungsbeziehungen zwischen zwei aufeinanderfolgenden Prozessschritten A und B (z. B. Ende-Anfang: B kann begonnen werden, sobald A beendet wurde)
- Festlegen einer zeitlichen Überlappungsbeziehung zwischen den Prozessschritten A und B (z. B. Ende-Anfang Beziehung und zwischen beiden Prozessschritten eine Wartezeit von 10 Minuten)

- Festlegen einer prozentualen Überlappung zwischen den Prozessschritten
- Festlegen einer Überlappung aufgrund fertiggestellter Teile des Prozessschrittes A

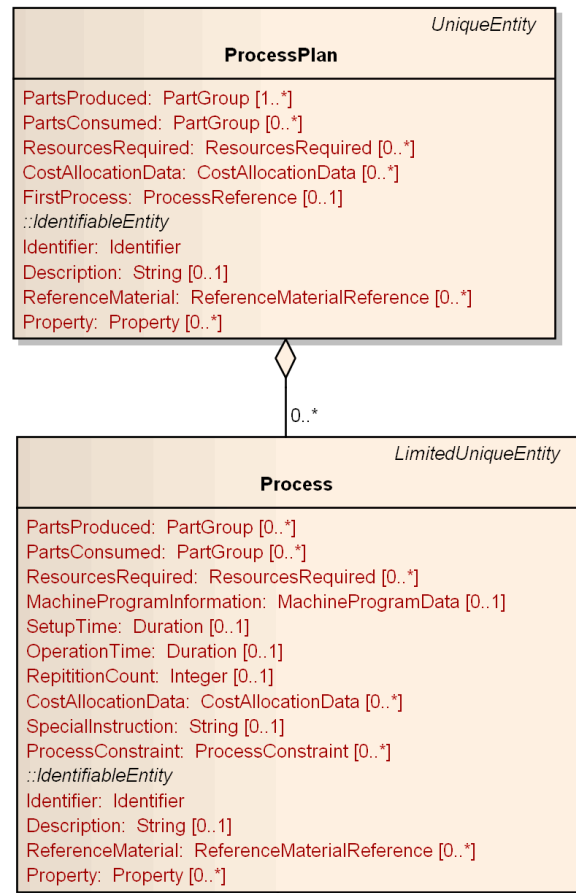


Abbildung 22: Ausschnitt der Klassendefinition zum Prozessplan [SISO10, S.133]

4.2.6 Resource Information Package

Im „Resource Information Package“ wurden Definitionen zur Abbildung von Ressourcen zusammengetragen. Hierunter sind sowohl Fertigungseinrichtungen (z. B. Maschinen, Förderanlagen) als auch Ausrüstungen (z. B. Spannvorrichtungen, Werkzeuge) und Mitarbeiter zu verstehen. Ressourcen können zu Klassen zusammengefasst werden, um eine Strukturierung aufgrund gleicher Charakteristiken zu ermöglichen [SISO10, S.144]. Abbildung 23 stellt die Klassendefinition der Ressource und -klasse dar.

Darüber hinaus ist es möglich, den Ressourcen einen vordefinierten Typ und Schichtpläne zuzuweisen sowie Referenzen zu anderen Ressourcen zu definieren, um damit auszudrücken, dass diese Ressourcen einen Arbeitsgang gemeinsam erledigen und in enger Verbindung zueinander stehen (z. B. Roboter mit Greifer). Weiterhin werden Strukturen zur Abbildung von Konfigurationszuständen von Maschinen angeboten. Mithilfe der Zusammenfassung von Ressourcen in einer Gruppe kann ermöglicht werden, dass diese nach außen hin als eine ein-

zige Ressource in Erscheinung treten. Dies fördert eine effiziente Zuordnung der Ressourcen zu den Produktionsoperationen. Innerhalb der Ressourcengruppe können Verbindungen zwischen Ressourcen festgelegt werden, um einen logischen Fluss zwischen den Elementen aufzuzeigen (vgl. ebenda S. 145). Für Mitarbeiter lassen sich Fähigkeitsprofile für Arbeitsstationen festlegen.

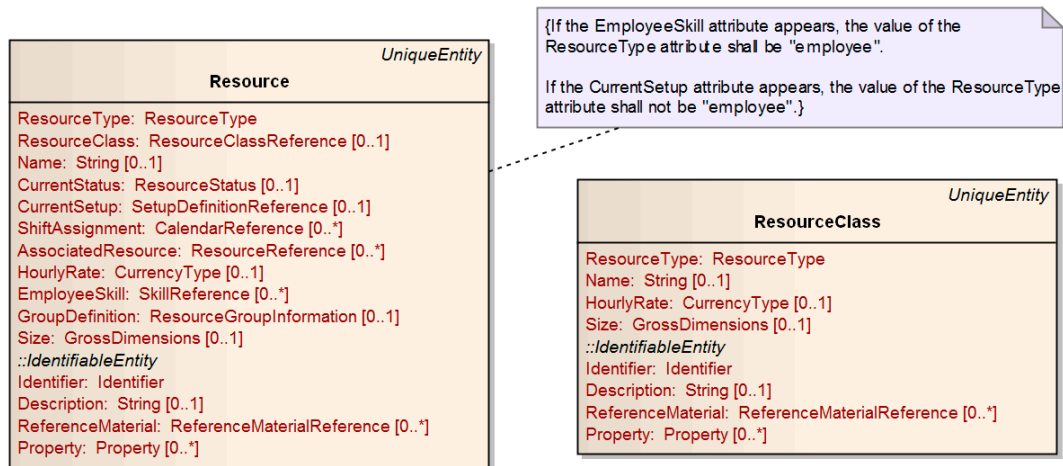


Abbildung 23: Klassendefinition zu Ressourcen und -klassen [SISO10, S.144]

Weitere Klassendiagramme des „Resource Information Package“ befinden sich im Anhang A.4.

4.3 Beitrag zur Integration automatisierter Modellgeneratoren

Ziel des Abschnittes ist es, zu beleuchten, welchen Beitrag das CMSD Information Model zur Integration automatisierter Modellgeneratoren in das Umfeld betrieblicher Anwendungssysteme der Produktionsplanung und -steuerung leisten kann.

Ein zentrales Problem der Integration stellt die Anbindung der Simulation an andere Anwendungssysteme der oft heterogenen Systemumwelt dar. Hierbei ist die Entwicklung von speziell angepassten, proprietären Schnittstellen teuer und behindert den konsequenten Einsatz der Simulationstechnik [Jo07, S.1674]. Einige Autoren (Je07 und Ro09) begegneten diesem Problem in der jüngeren Vergangenheit dadurch, dass sie eigene Datenmodelle unter Verwendung des XML-Formates zum Datenaustausch wählten, verbunden mit der Hoffnung zur Bildung eines standardisierten Modells beitragen zu können. Das XML-Format wird insbesondere wegen seiner Systemunabhängigkeit und der Möglichkeit, Daten in strukturierter Form abzulegen, als besonders geeignet für diesen Einsatzzweck angesehen [Ro09, S.121]. Des Weiteren kann es lizenzfrei genutzt werden.

Einen Eindruck inwieweit das CMSD Information Model zur Integration beitragen kann, soll nachfolgend anhand der Möglichkeiten zur Abbildung statischer und dynamischer Zusammenhänge von Fertigungssystemen gewonnen werden.

4.3.1 Abbildung statischer Zusammenhänge

Unter statischen Zusammenhängen sollen Informationen über Entitäten, die für Produktionsaktivitäten notwendig sind, aber keine Strategien oder Abläufe beschreiben, subsumiert werden. Darunter fallen jegliche Attribute der Entitäten, da diese nur Ausprägungen von Merkmalen festlegen, z. B. Eigenschaften der Fertigungseinrichtungen, Charakteristika von Teilen und Produkten. Weiterhin sollen u. a. auch Ablauf-, Prozess-, Schicht- und Wartungspläne dazu gezählt werden.

Im CMSD Information Model realisierte Datenstrukturen zur Abbildung statischer Zusammenhänge können als gelungen bezeichnet werden. Die Abdeckung simulationsrelevanter Informationen durch vordefinierte Strukturen ist sehr umfangreich, sinnvoll aufgebaut und recht einfach ergänzbar. Mittels des Property-Konzepts (vgl. Kapitel 4.2.2) wird gewährleistet, dass individuelle Erweiterungen der Klassen um Attribute und Verweise zu anderen Entitäten vorgenommen werden können. Hierdurch ist es ebenfalls möglich, Ergebnisdaten der Simulation in Entitäten des Fertigungssystems zu hinterlegen, z. B. Durchlaufzeiten und Terminüberschreitungen der Aufträge. Ein Aspekt, der dabei allerdings beachtet werden muss, ist, dass Ergebnisdaten der Simulation oft nur im Kontext sinnvoll interpretiert werden können. So ist z. B. die Angabe der Maschinenauslastung kein statisches Merkmal, sondern eine sich dynamisch ändernde Größe, die Ergebnis veränderlichen Systemsverhaltens ist. Eine Interpretation ist nur unter Einbeziehung des Kontextes möglich, u. a. die eingelasteten Aufträge und die Anwendung bestimmter Strategien und Abläufe. Dennoch kann es sinnvoll sein, Ergebnisdaten in Entitäten zu hinterlegen und hiermit anderen Anwendungssystemen leicht zugänglich zu machen.

Der Standard erfuhr über die Zeit hinweg stetige und spürbare Weiterentwicklungen. Dies trug zu einer erheblichen Verbesserung der Übersichtlichkeit und Leistungsfähigkeit bei. Unter anderem dadurch, dass die „Property“ eingeführt wurde und hierdurch auf eine Vielzahl von starren, vordefinierten Attributen verzichtet werden konnte. Auf der anderen Seite bewirkte das jedoch eine Steigerung der Eigenverantwortung der Anwender, nicht festgelegte Strukturen nachzumodellieren. Weiterhin ist es nun schwieriger zu beurteilen, ob Sachverhalte mittels des CMSD Information Model prinzipiell modellierbar sind. Hierbei wird es hilfreich sein, weitere Pilotprojekte mit Unterstützung industrieller Anwender umzusetzen, um in der Praxis festzustellen, wo die Grenzen des Standards aktuell noch erreicht werden und an welchen Stellen zusätzliche Strukturen oder Veränderungen sinnvoll erscheinen. Auf eine reizvolle, untersuchenswerte Nutzungsmöglichkeit des Standards, die Online-Kopplung zur Produktionssteuerung, wird im Kapitel 6.2 ein Ausblick gegeben.

Mit der Integration von Strukturen zur Beschreibung von Geometrieinformationen im „Layout Package“ wurde der Standard konsequent erweitert. Nun stehen Modellierungsmöglichkeiten zur Verfügung, mit denen Layoutdaten der Fertigungseinrichtungen auf einfache Weise in die Betrachtung einfließen können.

Im Kapitel 5 werden konkrete Modellierungsvorschläge für die Datenstrukturen eines Beispielszenarios unter Einbeziehung der flexiblen Abbildungsmöglichkeiten gegeben.

4.3.2 Abbildung dynamischer Zusammenhänge

Unter dynamischen Zusammenhängen sollen Strategien und Abläufe der Produktionssteuerung zusammengefasst werden (vgl. Kapitel 2.3.5).

Die Entwicklung des CMSD Information Model verfolgt das Ziel, simulationsrelevante Informationen in standardisierten Datenstrukturen für den Datenaustausch bereitzustellen. Jedoch ist es nicht dazu gedacht, eine vollständige externe Repräsentation sämtlicher, notwendiger Informationen zur Abbildung von Simulationsmodellen zu ermöglichen. Insbesondere bietet es keine Beschreibungsmöglichkeiten zur Abbildung von Logiken und algorithmischen Formulierungen. Strategien und Abläufe der Produktionssteuerung geben Handlungsbeschreibungen wieder, um übergeordnete Ziele, wie z. B. hohe Maschinenauslastungen oder kurze Durchlaufzeiten, zu erreichen.

Für Beschreibungsmethodiken zur Abbildung von Strategien und Abläufen nennt Selke u. a. die Anforderung, Ablaufsequenzen und Entscheidungssituationen beschreiben zu können [Se05, S.52f]. Diese Anforderung kann augenblicklich nicht durch das CMSD Information Model erfüllt werden. Aus dem Grund ist es derzeit nicht möglich, eine im Standard verankerte, formalisierte Beschreibung von Strategien und Abläufen der Produktionssteuerung vorzunehmen und zu transportieren.

Dennoch lassen sich Möglichkeiten finden, Strategien und Abläufe im Simulationsmodell zu berücksichtigen und hierfür Informationen über CMSD-Dokumente auszutauschen. Eine einfache und pragmatische Lösung ist, die Strategien in der Simulationsapplikation vordefiniert in Bausteinen zur Verfügung zu stellen und per Datenaustausch die Bezeichnungen zu transportieren. Das setzt allerdings voraus, dass die angewendeten Strategien bekannt sein müssen. Erfahrungen aus der Praxis zeigen jedoch, dass auch Strategien zur Anwendung kommen, die nicht in rechnergestützten Systemen gepflegt werden [Je07, S.107].

Als weitere Möglichkeit könnte eine Vorgehensweise in Betracht gezogen werden, die dem Ansatz von Selke [Se05] zur Interpretation von Strategien und Abläufen aus BDE-Daten entlehnt ist. Hiernach könnten der Simulationsanwendung per CMSD-Dokumenten Daten aus einem BDE oder PPS-System für die Interpretation von Strategien zur Verfügung gestellt werden. Leistungsfähige Simulationsanwendungen bieten oft umfangreiche Möglichkeiten zur individuellen Erweiterung mithilfe integrierter Programmiersprachen. Die eigentlichen Strategiebeschreibungen müssten auch in diesem Fall schon vordefiniert in Bausteinen bereitgestellt werden, wobei die Bausteine nach einer Strategieinterpretation ausgewählt, kombiniert und ggf. parametrisiert werden könnten. Ob die notwendigen Eingangsdaten jedoch mittels des CMSD Information Model hinreichend abgebildet werden können, müsste vorab genau betrachtet werden.

4.4 Fazit

Der aktuelle Stand des CMSD Information Modell kann insgesamt als vielversprechend angesehen werden. Es bietet sehr umfassende und gut durchdachte Strukturen zur Abbildung simulationsrelevanter Daten. Mit der Auswahl des XML-Datenformates wurde auf eine zukunftssträchtige Technologie gesetzt, die viel Freiraum für zusätzliche Erweiterungen und Anpassungen lässt. Weitere Vorteile des Formates sind in der Systemunabhängigkeit, Offenheit und Flexibilität zu sehen, welche im heutigen Umfeld heterogener Systemlandschaften vollends zum Tragen kommen.

Dem Hauptziel des Standards, das Ermöglichen eines effizienten Datenaustausches zwischen betrieblichen Anwendungssystemen und Simulationsanwendungen durch Bereitstellen eines leistungsfähigen Datenformates, konnte ein gutes Stück näher gekommen werden. Das CMSD Information Model scheint hierbei einige andere Ansätze der letzten Jahre (vgl. Je07 und Ro09) deutlich zu übertreffen.

Ein Nachteil, der mit den flexiblen und nicht gänzlich restriktiven Datenstrukturen (siehe Kapitel 4.2.2 zum Konzept der „Property“) eingegangen wurde, ist die Möglichkeit, viele unterschiedliche konkrete Umsetzungen der Strukturen über gleiche Sachverhalte formulieren zu können. Unterschiedliche Bedürfnisse oder Herangehensweisen der Anwender könnten dazu führen, dass die Einigung auf einheitliche Datenstrukturen erschwert wird. Diese werden aber notwendig sein, um homogene Datenanbindungen an die entsprechenden Anwendungssysteme entwerfen zu können. Eine Möglichkeit diesen Nachteil abzufedern, ist die Erarbeitung umfassender und an der Praxis ausgerichteter Referenzstrukturen zur aktuellen Version des Standards, welche zur öffentlichen Diskussion gestellt werden. Das CMSD Information Model scheint seit einiger Zeit eine ausreichende inhaltliche Kontinuität zu haben, um das Ausmaß von Anpassungen der Referenzstrukturen zu begrenzen. Außerdem würde es dazu beitragen, die Aufmerksamkeit der Fachwelt für den Standard zu erhöhen und den Initialaufwand bei Nutzung deutlich zu reduzieren.

5 Prototypische Umsetzung eines automatisierten Modellgenerators

Im vorliegenden Kapitel wird eine prototypische Umsetzung eines automatisierten Modellgenerators vorgestellt. Dies geschieht anhand eines Beispielszenarios einer Werkstattfertigung¹² unter Einbeziehung von einfachen Steuerstrategien zur Reihenfolgegestaltung und Ressourcenbelegung (vgl. Tabelle 1 in Kapitel 2.3.5). Die erforderlichen Eingangsdaten werden dem Modellgenerator im CMSD-Datenformat der Version vom April 2010 übergeben. Erläuterungen zu wesentlichen Eingangsdatenstrukturen sind im Abschnitt 5.4 zu finden.

Durch die prototypische Umsetzung soll einerseits die Leistungsfähigkeit des Standards im begrenzten Rahmen getestet und andererseits eine Ausgangsbasis für weitere Forschungsarbeiten geschaffen werden. Das Ziel ist eine Simulationsanwendung, die sich sukzessiv erweitern lässt und einen tieferen Einblick in die Nutzbarkeit des CMSD Information Model für den Datenaustausch zwischen betrieblichen Anwendungssystemen einschließlich der Simulation erlaubt.

5.1 Integration in das Umfeld betrieblicher Anwendungssysteme

Zunächst soll eine mögliche Integrationslösung vorgestellt werden, die zeigt, wie der Modellgenerator prinzipiell in die Systemlandschaft eingebunden sein könnte. Der hierfür notwendige Funktionsumfang der Anwendung kann aufgrund des großen Aufwandes in dieser Arbeit nicht erreicht werden. Dennoch ließe sich der vorliegende Ansatz als Ausgangspunkt für zusätzliche Erweiterungen nutzen, um zu einer umfassenden Applikationslösung zu gelangen.

Abbildung 24 zeigt die Simulationsanwendung als integrierten Bestandteil zwischen Systemen der Fertigungsplanung, der Produktionsplanung und -steuerung und dem Fertigungsmanagement¹³. Von der rechnergestützten Prozessplanung würden Eingangsdaten, welche das Fertigungskonzept¹⁴ beschreiben, für den Modellgenerator bereitgestellt. Dies sind im

12 In Werkstattfertigungen erfolgt die Anordnung der Arbeitsstationen nach dem Verrichtungsprinzip. Das bedeutet, gleiche oder ähnliche Tätigkeiten (Verrichtungen) werden räumlich zu einer Gruppe (Werkstatt) zusammengefasst.

13 Das Fertigungsmanagement ist unterhalb der Ebene des Unternehmensmanagements (ERP / PPS) und oberhalb der Fertigungsebene angesiedelt. Es unterstützt u. a. die Feinplanung und -steuerung indem, ausgehend von Produktionszielen, Arbeitsvorräten, Ressourcen- und Materialinformationen, ein Reihenfolge- und Belegungsplan der Aufträge bzw. Arbeitsgänge erstellt wird [VDI 5600, S.16]. Weitere Aufgaben sind das Informations-, Betriebsmittel-, Personal-, Qualitäts- und Materialmanagement sowie die Leistungsanalyse und Datenerfassung (vgl. ebenda S. 7). IT-Systeme, die zur Unterstützung des Fertigungsmanagements eingesetzt werden, heißen Manufacturing Execution Systems (MES).

14 Das Fertigungskonzept beschreibt, was, wie und mithilfe welcher Ressourcen gefertigt werden soll. Es führt hierzu die Elemente Produkt, dessen Verbindungsdaten der Einzelteile, Prozessbeschreibung und Ressource zusammen. Zur Erstellung des Fertigungskonzeptes werden im Rahmen der Digitalen Fabrik rechnergestützte Planungstools bereitgehalten.

Beispielszenario Informationen zu den Teilen, Ressourcen (z. B. Positionsdaten und Puffergrößen) und die Prozessbeschreibung.

Alternativ könnte eine Lösung in Betracht gezogen werden, bei der diese Daten weitestgehend das ERP-System zur Verfügung stellt. Fehlende Informationen über z. B. Positionsdaten der Ressourcen oder Materialflusselemente, wie Quellen und Senken, müssten daraufhin jedoch manuell eingegeben bzw. durch Algorithmen im Modellgenerator kompensiert werden. Denkbar wäre hierbei der Einsatz von Heuristiken, welche die Anordnung der Elemente im Layout unterstützen und Vorgehensweisen, welche basierend auf dem Arbeitsplan auf Quellen und Senken des Fertigungssystems schließen.

Aus dem ERP-System können darüber hinaus Daten zur Arbeitsorganisation (z. B. Schichtmodelle) und Stördaten der Ressourcen gewonnen werden. Die Stördaten sind ggf. aufzubereiten, um sie in komprimierter Form im Simulationsmodell einsetzen zu können.

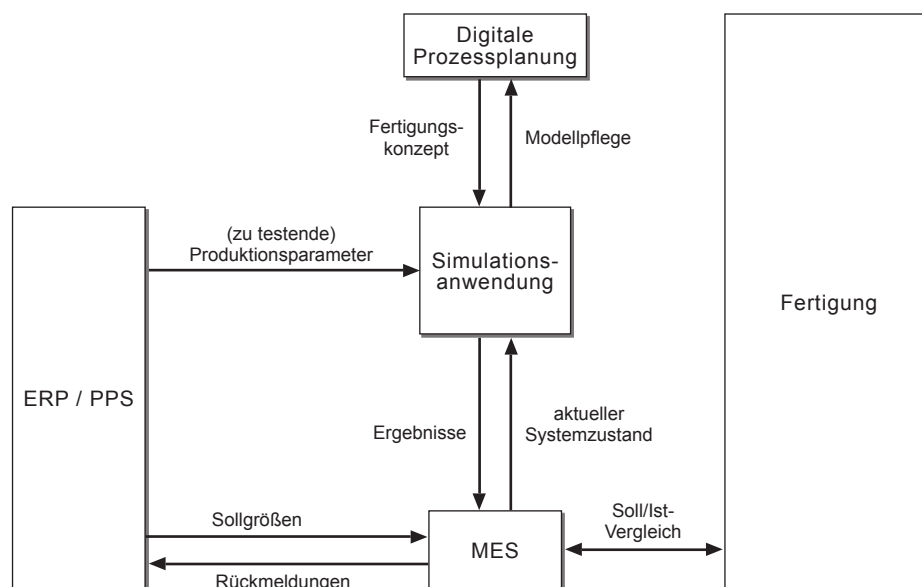


Abbildung 24: Integration der Simulationsanwendung in das betriebliche Umfeld

Ausgehend von diesem planungsbegleitenden Ansatz wäre es denkbar, Potenziale beim betriebsbegleitenden Einsatz zu erschließen. Hierfür wird zusätzlich die Kopplung an ein MES-System vorgeschlagen. Dieses führt einen permanenten Soll/Ist-Vergleich zwischen geplanter und tatsächlicher Situation durch und ermöglicht die Übertragung des aktuellen Zustandes des Fertigungssystems an die Simulationsanwendung. Dadurch kann gewährleistet werden, dass sich das Simulationsmodell zu Beginn der Experimentierphase in einem, dem realen Fertigungssystem möglichst nahekommenden Zustand befindet. Die zu testenden Produktionsparameter (z. B. Produktionsprogramm) übermittelt das ERP-System an die Simulation, welche die Ergebnisse der Experimente zum Fertigungsmanagement weiterleitet. Dort werden unter Einbeziehung der Simulationsergebnisse Rückschlüsse bezüglich der zutreffenden Steuerungsparameter gezogen und entsprechende Maßnahmen veranlasst.

5.2 Struktur des Modellgenerators

Als Simulationswerkzeug fiel die Wahl auf Plant Simulation von der Firma Siemens PLM Software. Die Gründe hierfür sind seine Bausteinorientierung und die Möglichkeit, umfassende Erweiterungen und Logiken mittels der eingebetteten Programmiersprache SimTalk zu programmieren. Des Weiteren stellt Plant Simulation eine XML-Schnittstelle bereit, welche für die Umsetzung von zentraler Bedeutung ist.

In Abbildung 25 ist das Gesamtkonzept des Modellgenerators und seines Umfeldes dargestellt. Wie bereits angedeutet, konnte im Rahmen der vorliegenden Arbeit nur der Grundstein für zukünftige Betrachtungen gelegt werden. Den Schwerpunkt bildet die Erarbeitung eines Konzeptes, das erweiterbar und zu einer integrierten Lösung ausbaufähig ist.

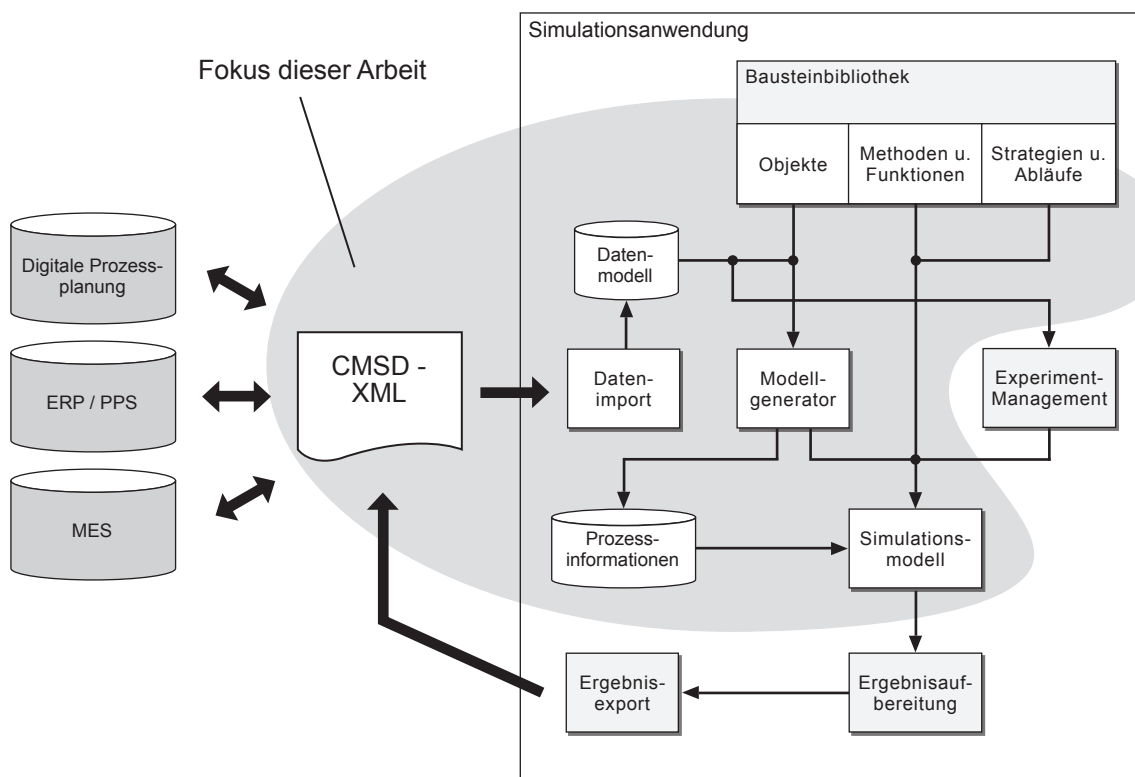


Abbildung 25: Gesamtkonzept des Modellgenerators in Anlehnung an [Ro09, S.147]

Die Betrachtung setzt bei der Erstellung der XML-Eingabedatei im CMSD-Format ein. Diese wurde unter Verwendung eines XML-Editors (XMLSpy der Firma Altova) aus den zur Verfügung stehenden Schemata¹⁵ erzeugt und enthält alle Eingabedaten, die dem Modellgenerator automatisiert bereitgestellt werden. Eine Beschreibung der Bestandteile des Generators (vgl. Abbildung 26) wird in den nachfolgenden Abschnitten gegeben.

¹⁵ Eine Sammlung der CMSD-Schemata ist unter folgender Seite abrufbar (Stand 07.09.2010): <http://simrest.svn.sourceforge.net/viewvc/simrest/CMSD/branches/fromUML/utis/Schema%20for%20CMSD%20V3b>.

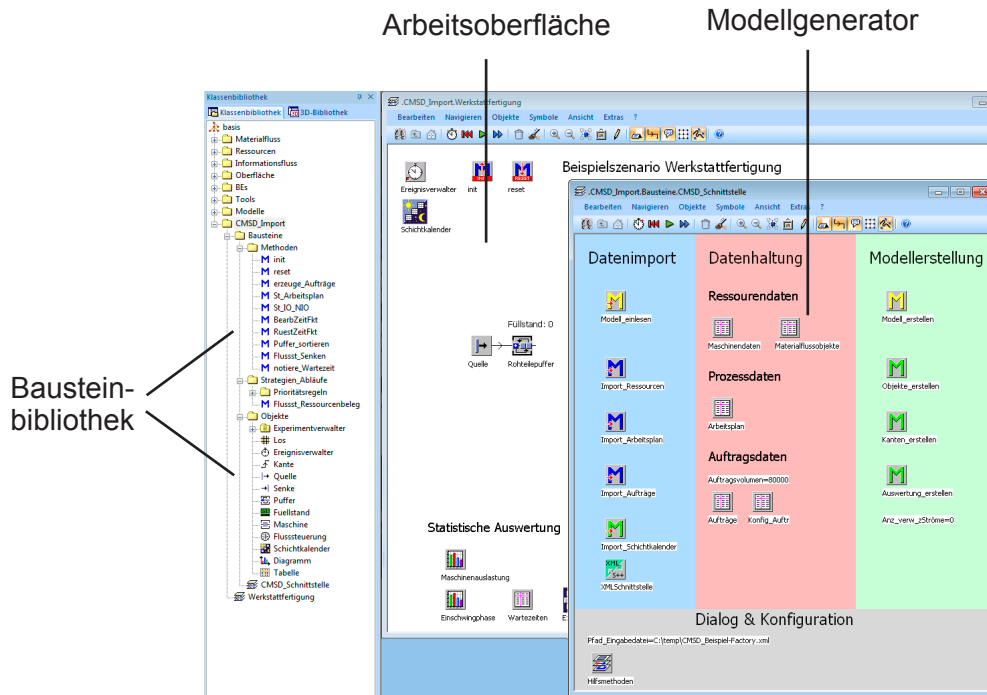


Abbildung 26: Übersicht der Bestandteile des Modellgenerators

Der Modellgenerator ist in die vier Funktionsbereiche Datenimport, Datenhaltung, Modellerstellung und Dialog & Konfiguration gegliedert. Letzteres konnte bis zum Abschluss dieser Arbeit nicht vollends entwickelt werden. Prinzipiell soll dieser Bereich aber dazu dienen, ein Dialogsystem bereitzustellen mit deren Hilfe der Anwender durch die Programmfunktionen geführt wird. Somit verfolgt er das Ziel einer Verbesserung der Bedienbarkeit. Weiterhin können in ihm allgemeine Variablen und Hilfsmethoden, welche Dienstleisterfunktionen für andere Methoden des Generators übernehmen, abgelegt werden.

5.2.1 Bausteinbibliothek

Der Modellgenerator stützt sich auf ein Bibliothekskonzept, welches alle im Modell verwendbaren Objekte und Funktionen enthält. Strukturiert ist die Bibliothek in drei grundsätzliche Bereiche:

Objekte

- Materialflusselemente (z. B. Quelle, Senke, Maschine und Puffer)
- Elemente zur Überwachung, Auswertung und Experimentverwaltung (Display, Diagramm und Experimentverwalter)
- Weitere Objekte (z. B. Auftragslos und Schichtkalender)

Methoden und Funktionen

- Methoden zur Steuerung der Simulation (Ablauf und Verteilen der Aufträge auf die Senken)
- Methoden und Funktionen zur Bereitstellung von Eingabeparametern (Initialisierung des Modells, Erzeugen der Aufträge, Bereitstellen der Bearbeitungs- und Rüstzeiten)
- Methoden, die Servicefunktionen übernehmen (Notieren von Statistikdaten, Sortierfunktionen und Erzeugen von schadhaften Teilen)

Strategien und Abläufe

- Prioritätsregeln zur Reihenfolgegestaltung (exemplarische Implementierung der Regel KOZ)
- Strategien zur Ressourcenauswahl (Implementierung der Regel: Zuteilung derjenigen Maschine, welche die geringste Bearbeitungszeit bietet)

Für den Bereich der „Objekte“ wurde auf Standardbausteine des Simulators zurückgegriffen. Einige der Bausteine (z. B. das Element Maschine) sind bereits parametrisiert und mit Funktionsaufrufen zur Ermittlung von Eingangsdaten sowie mit Aufrufen von Servicemethoden versehen. Zukünftige Erweiterungen, insbesondere individuell erstellte Objekte, können an dieser Stelle verwaltet werden.

5.2.2 Datenimport

Die automatisierte Dateneingabe ist auf vier Methoden mit den Bereichen Ressourcen, Arbeitsplan, Aufträge und Schichtkalender aufgeteilt. Eingelesen werden die Daten aus einer CMSD-XML-Datei, die einen Großteil der benötigten Eingangsdaten bereithält. Für den Import war es möglich, die XML-Schnittstelle von Plant Simulation einzusetzen und somit ein Zurückgreifen auf die Unterstützung der Abfragesprache XML Path Language (XPath) beim Einlesen zu nutzen.

Eine vollständige Automatisierung der Datenversorgung des Modellgenerators wird im Prototypen nicht erreicht. Das liegt unter anderem daran, dass mittels manueller Erstellung der Eingangsdaten keine bedeutende Anzahl von Fertigungsaufträgen bereitgestellt werden kann. Die Aufträge werden deshalb im Simulator basierend auf empirischen Verteilungen der Auftragsstypen und ihrer Losgrößen erstellt. Hierfür sind jedoch Informationen über die auftretenden Losgrößen und deren Häufigkeiten erforderlich. Diese müssen manuell in eine Tabelle eingepflegt werden, da im betrachteten Szenario angenommen wurde, dass keine automatische Aufbereitung der Betriebsdaten stattfindet, welche die Verteilungen ermittelt. Die Abbildung von empirischen Verteilungen der Aufträge ist im CMSD-Format prinzipiell möglich. Jedoch ist es hierbei notwendig, die statistischen Auftragsbeschreibungen von den

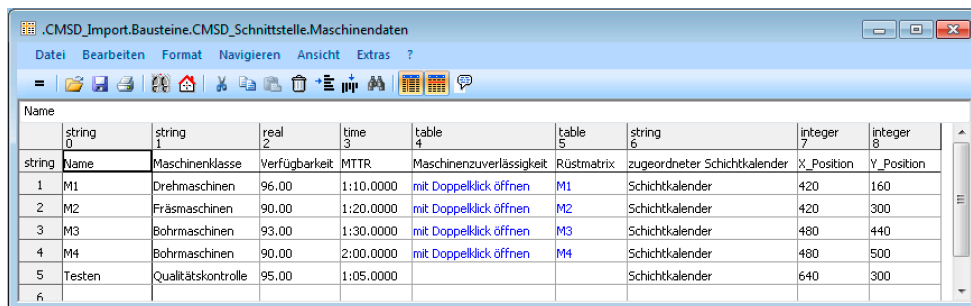
Beschreibungen aktuell ausgeführter oder geplanter Aufträge zu unterscheiden. Der Entitätstyp „Job“ würde demnach einmal zur Abbildung aktueller bzw. zukünftiger Aufträge genutzt werden und zum anderen zur Formulierung stochastischer Verteilungen der Auftragsarten.

Weiterhin sind die zur Durchführung von Simulationsexperimenten notwendigen Experimentdaten, wie z. B. Versuchsdauer, Einschwingdauer und Versuchsanzahl, manuell einzugeben, da nicht davon ausgegangen werden kann, dass diese Daten aus dem Umfeld der betrieblichen Anwendungssysteme des betrachteten Szenarios ausgelesen werden können.

Die Importmethoden zum Einlesen des Arbeitsplanes, der Ressourcen und Materialflussobjekte sowie des Schichtkalenders sind im Anhang B.1 bis B.3 zu finden.

5.2.3 Datenhaltung

Durch Importmethoden eingelesene Daten werden in der Regel in Tabellen des Modellgenerators verwaltet. Nur bei den Angaben zum Schichtkalender wird hiervon abgewichen, um sie im Baustein Schichtkalender zu hinterlegen. Dieser ist in Plant Simulation als Standardbaustein enthalten. Abbildung 27 zeigt exemplarisch die Datentabelle zu den Maschinendaten.



	Name	string 0	string 1	real 2	time 3	table 4	table 5	string 6	integer 7	integer 8
string	Name		Maschinenklasse	Verfügbarkeit	MTTR	Maschinenzuverlässigkeit	Rüstmatrix	zugeordneter Schichtkalender	X_Position	Y_Position
1	M1		Drehmaschinen	96.00	1:10.0000	mit Doppelklick öffnen	M1	Schichtkalender	420	160
2	M2		Fräsmaschinen	90.00	1:20.0000	mit Doppelklick öffnen	M2	Schichtkalender	420	300
3	M3		Bohrmaschinen	93.00	1:30.0000	mit Doppelklick öffnen	M3	Schichtkalender	480	440
4	M4		Bohrmaschinen	90.00	2:00.0000	mit Doppelklick öffnen	M4	Schichtkalender	480	500
5	Testen		Qualitätskontrolle	95.00	1:05.0000			Schichtkalender	640	300
6										

Abbildung 27: Datentabelle der Maschinendaten

Die Verwaltung der Daten in Tabellen vereinfacht es für den Anwender, einen Überblick zu den Angaben der Objekte zu gewinnen sowie Änderungen vorzunehmen. Ein weiterer Vorteil der zentralen Datenhaltung ist die übersichtliche Bereitstellung zu variierender Eingangsparameter für die Experimentdurchführung mittels des Experimentverwalters.

5.2.4 Modellerstellung

Nach dem Einlesen der Daten kann der Modellaufbau durch den Anwender initiiert werden. Hierfür steht eine Methode zur Verfügung, welche durch Aufrufe jeweils spezialisierter Methoden die einzelnen Schritte des Modellaufbaus einleitet. Für zukünftige Erweiterungen wird empfohlen, den Modellaufbau über ein Dialogsystem zu starten.

Beim Erstellen des Modells werden basierend auf der Datengrundlage der Datentabellen zunächst Instanzen der Objekte im Layout der Arbeitsoberfläche platziert und anschließend parametrisiert. Zu den Positionsdaten der Maschinen, Puffer sowie der Quellen und Senken wurde eine Vereinfachung gewählt, die darin besteht, dass die XML-Eingabedatei bereits Zielkoordinaten der Objekte auf der Arbeitsoberfläche in Plant Simulation bereitstellt. Auf eine Formulierung zusätzlicher Koordinatendaten der Elemente mit anschließender Transformation konnte deswegen verzichtet werden.

Auf die Erstellung und Parametrisierung der Instanzen folgt das Setzen von Verbindungen. Das geschieht anhand der Vorgänger- und Nachfolgerbeziehungen der Materialflussobjekte, die aus einer Datentabelle entnommen werden. In der CMSD-Eingabedatei sind hierfür Verbindungen zwischen den Ressourcen, z. B. Zuordnung von Eingangs- und Ausgangspuffern zu den Maschinen, hinterlegt. Nachdem die Verbindungen zunächst so gesetzt wurden, wie sie in der Eingabedatei formuliert sind, erfolgt daraufhin das Einfügen einer Flusssteuerung vor den Senken und innerhalb von Maschinengruppen, die aus mehreren Maschinen bestehen. Die Flusssteuerungen dienen dazu, die Aufträge auf die entsprechenden Senken zu verteilen und eine Ressourcenauswahl zu ermöglichen, falls ein Arbeitsgang eines Auftrags auf mehreren Maschinen einer Gruppe erledigt werden kann. Zum Baustein Flusssteuerung befindet sich in der CMSD-Eingabedatei kein Gegenstück, da in den angeschlossenen betrieblichen Anwendungssystemen des betrachteten Szenarios, insbesondere bei der digitalen Prozessplanung, standardmäßig keine Entsprechung dieses Bausteins bereitgestellt wird. Die Flusssteuerung wird aufgrund dessen im Modellgenerator regelbasiert eingefügt.

Zum Schluss der Routinen zur Modellerstellung werden beispielhaft einige Objekte erstellt, welche für statistische Betrachtungen nutzbar sind. Mithilfe dieser wird die Maschinenauslastung während eines Simulationslaufes dargestellt und eine grafische Unterstützung zur Ermittlung der Einschwingdauer gegeben. Bei späteren Entwicklungen ist es anzustreben, diese auszubauen, weiterzuentwickeln und in ein Experimentmanagement einzubinden.

5.3 Anwendungsszenario

Nachfolgend wird das gewählte Anwendungsbeispiel beschrieben auf dessen Grundlage die CMSD-Eingangsdaten formuliert wurden und das es ermöglicht, den Modellgenerator exemplarisch zu testen.

Als Beispiel wurde ein Szenario einer fiktiven Werkstattfertigung zur mechanischen Bearbeitung gewählt (siehe Abbildung 28). In diesem werden drei Produkte in Losfertigung hergestellt, wobei keine Montagearbeiten stattfinden. Der Maschinenpark setzt sich aus vier Bearbeitungsstationen und einer Station zur Qualitätssicherung zusammen.

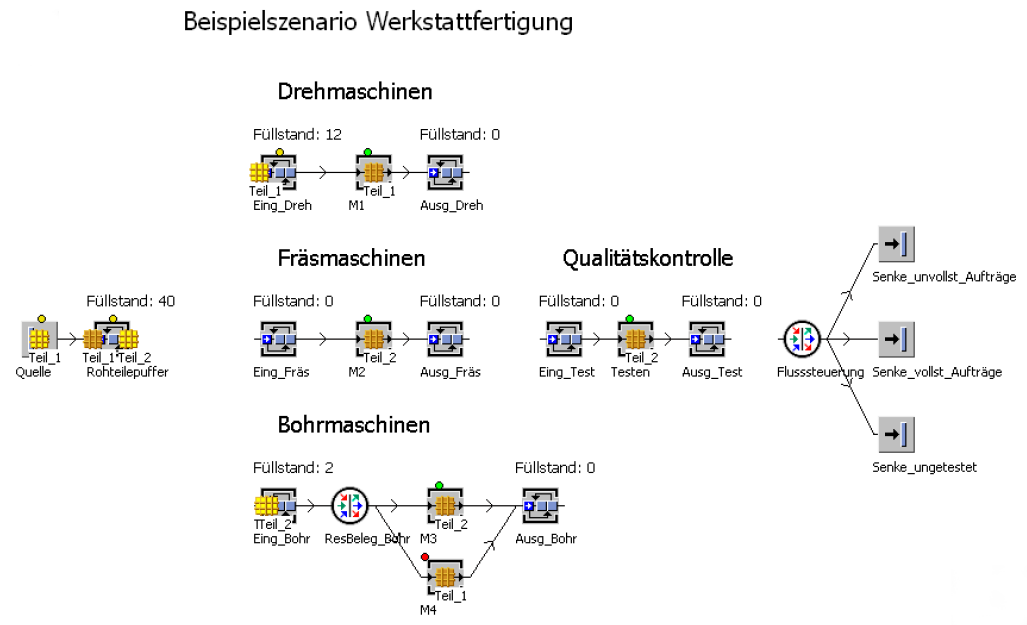


Abbildung 28: Anwendungsbeispiel für den Modellgenerator

Die Maschinen sind nach dem Verrichtungsprinzip drei Maschinengruppen (Werkstätten) zugeordnet und besitzen, ebenso wie die Qualitätssicherung, jeweils eine Bereitstellungs- und Abholefläche (Eingangs- und Ausgangspuffer) mit festgelegten Kapazitäten. Weiterhin existiert ein zentraler Rohteilepuffer von dem aus die Arbeitsstationen mit den Rohteilen versorgt werden.

Nach der Qualitätssicherung erfolgt eine Verteilung der Auftragslose auf die Senken, wobei die Einteilung danach vorgenommen wird, ob ein Auftrag vollständig ist, d. h. ohne schadhafte Teile gefertigt wurde, nicht vollständig ist bzw. nicht getestet wurde. Im Szenario durchlaufen alle Auftragslose die Qualitätssicherung solange der Arbeitsgang im Arbeitsplan aufgeführt wird. Der Testvorgang ließe sich auch so interpretieren, dass nicht alle Teile eines Loses tatsächlich geprüft werden, sondern, dass das Testen einer Stichprobe durchschnittlich eine bestimmte Zeit pro Teil des Auftragsloses in Anspruch nimmt.

Im Anwendungsbeispiel kommen die beiden zusätzlich implementierten Strategien der Produktionssteuerung zum Einsatz, die Prioritätsregel KOZ in den Eingangspuffern der Maschinengruppen und die Strategie zur Ressourcenauswahl in der Werkstatt der Bohrmaschinen. Für das Szenario mussten eine Reihe von Einschränkungen vorgenommen werden, um den Implementierungsaufwand zu begrenzen. Keine Berücksichtigung fanden u. a. Transportvorgänge und -zeiten, Terminvorgaben bei der Fertigung sowie Nacharbeiten oder erneutes Produzieren bei schadhaft gefertigten Teilen.

5.4 Struktur der CMSD-Eingangsdaten

Für das zuvor beschriebene Anwendungsbeispiel wurde eine auf dem Standard CMSD basierende XML-Eingabedatei angelegt. In Abbildung 29 ist der Grundaufbau von CMSD-Dokumenten dargestellt. Diese bestehen aus einem Datenkopf, in welchem u. a. Metadaten,

Festlegungen zu Standardeinheiten und Referenzen zu externen Dokumenten hinterlegt werden können. In der Datensektion werden die Datenstrukturen zu den Entitäten aufgenommen, welche das Fertigungssystem abbilden.

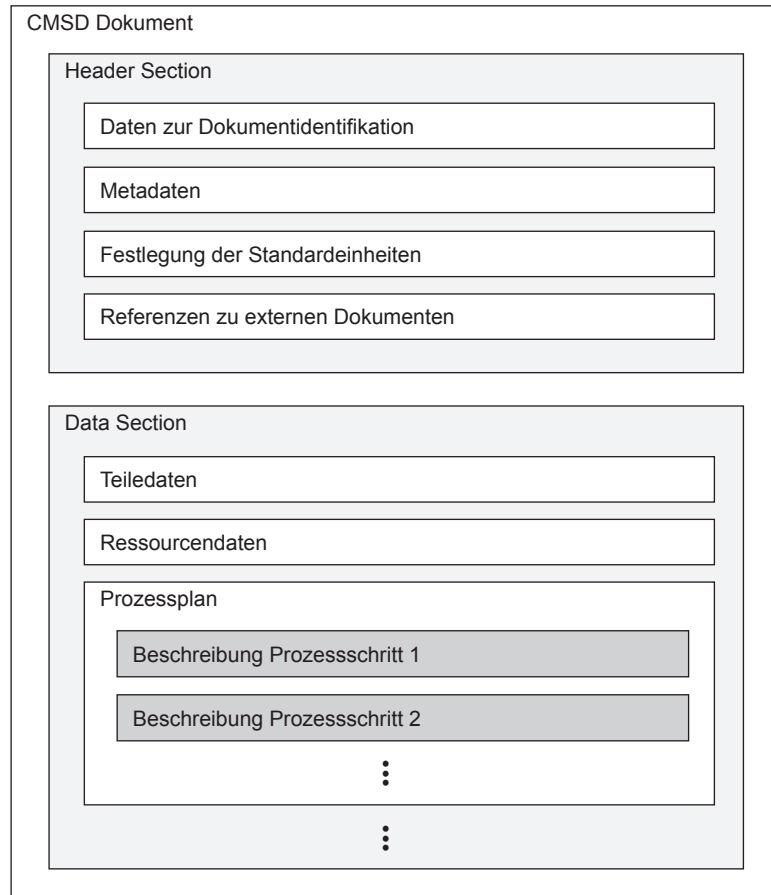


Abbildung 29: Grundaufbau eines CMSD-Dokuments [SISO10, 19]

In den darauffolgenden Abschnitten soll auf wesentliche, konkrete Beschreibungen der CMSD-Eingangsdaten eingegangen werden. Ein Augenmerk wird dabei auf der Nutzung der flexiblen Abbildungsmöglichkeiten des CMSD-Standards liegen.

5.4.1 Layout Package

Das Layout Paket kam zur Hinterlegung der Positionsdaten der Maschinen, Bereitstell- und Abholflächen (Auftragspuffer) aber auch der Materialflussobjekte Quelle und Senke, was eine Vereinfachung darstellt, zum Einsatz. Dem CMSD-Standard entsprechend, wurde hierfür ein Fertigungsbereich definiert, in dem sich die Elemente befinden. Innerhalb einer Instanz „Layout“ wurden die Positionsdaten der Elemente in den „Platzierungen“ (Placement) abgelegt. Für jede dieser „Platzierung“ existiert eine „LayoutObject“ Instanz, welche die Verbindung zum jeweiligen Ressourcen- bzw. Materialflussobjekt herstellt. Ein Auszug der Datendefinitionen ist in Abbildung 30 zu sehen.

```

<Layout>
  <Identifier>X Halle_1</Identifier>
  <Description>Produktionshalle 1 am Ort X</Description>
  <Name>Halle_1</Name>
  <Boundary>
    ...
  </Boundary>
  <Placement>
    <LayoutElementIdentifier>Lay_Werk01_M1</LayoutElementIdentifier>
    <Location>
      <X>420</X>
      <Y>160</Y>
    </Location>
  </Placement>
  ...
</Layout>

<LayoutObject>
  <Identifier>Lay_Werk01_M1</Identifier>
  <Description>Layoutinformationen für M1</Description>
  <Name>M1</Name>
  <AssociatedResource>
    <ResourceIdentifier>Werk01_M1</ResourceIdentifier>
  </AssociatedResource>
  <Boundary>
    ...
  </Boundary>
  <Property>
    <Name>Standort</Name>
    <Description>Standort des Layoutelements (X ist Ortsbezeichnung)</Description>
    <Unit>String</Unit>
    <Value>X_Halle_1</Value>
  </Property>
  <Type>machine</Type>
</LayoutObject>
    
```

Abbildung 30: Auszug der Layoutinformationen aus der CMSD-Eingabedatei in XML

5.4.2 Production Operations Package

Aus diesem Paket fand nur der Entitätstyp Auftrag (Job) Anwendung. Beispielhaft wurden zwei Aufträge formuliert, die im Anwendungsszenario genutzt werden können. Ein Ausschnitt der genutzten Datenstrukturen ist in Abbildung 31 dargestellt.

```

<Job>
  <Identifier>J1</Identifier>
  <PrecedenceConstraint>
    <PrecedenceRelationship>FS</PrecedenceRelationship>
    <LagPercentage>100</LagPercentage>
    <LagPartsComplete>1</LagPartsComplete>
    <SuccessorJob>
      <JobIdentifier>J2</JobIdentifier>
    </SuccessorJob>
  </PrecedenceConstraint>
  <PlannedEffort>
    <PartsProduced>
      <Description>Teil_2</Description>
      <PartType>
        <PartTypeIdentifier>Teil_2</PartTypeIdentifier>
      </PartType>
      <PartQuantity>90</PartQuantity>
    </PartsProduced>
    <PartsConsumed>
      <Description>Rohteil</Description>
      <PartType>
        <PartTypeIdentifier>Rohteil</PartTypeIdentifier>
      </PartType>
      <PartQuantity>90</PartQuantity>
    </PartsConsumed>
    <ProcessPlan>
      <ProcessPlanIdentifier>Prozessplan Teil_2</ProcessPlanIdentifier>
    </ProcessPlan>
  </PlannedEffort>
</Job>
    
```

Abbildung 31: Darstellung eines Auftrages aus der CMSD-Eingabedatei in XML

Für das beschriebene Anwendungsszenario genügte die Nutzung der im CMSD-Standard vordefinierten Strukturen, da nur die Angaben zum Teiletyp und der Losgröße relevant waren.

5.4.3 Production Planning Package

Zur Formulierung des Schichtkalenders und der Arbeitspläne kamen die Entitätstypen „Calendar“ und „ProcessPlan“ des „Production Planning“ Paketes zum Einsatz.

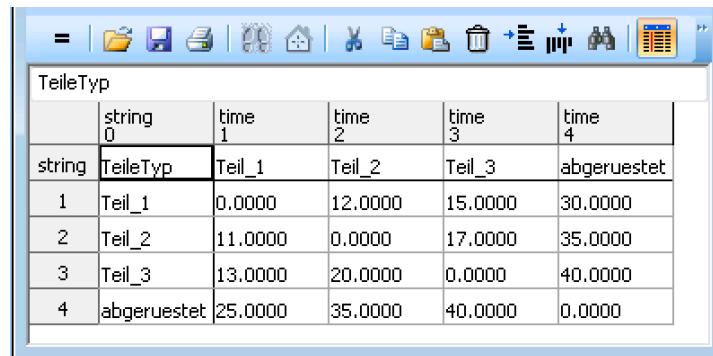
Im Anwendungsbeispiel besteht die Möglichkeit zur Verwendung einer Strategie zur Ressourcenauswahl. Um dies gewährleisten zu können, war es erforderlich, den Arbeitsplan so zu formulieren, dass die Bearbeitung des entsprechenden Arbeitsganges, bei dem die Strategie zur Anwendung kommt, auf mehreren Maschinen möglich ist. Innerhalb der Arbeitspläne wurden die Arbeitsgänge mittels des Entitätstyps „Process“ modelliert, wobei ein Arbeitsgang, der auf mehreren Maschinen durchführbar ist, als eigenständiger „Process“ (in der XML-Beschreibung als Prozessalternative bezeichnet) hinterlegt wurde. Durch den Datenimport bedingt, bestand die Notwendigkeit einen Verweis der Arbeitsfolge auf die entsprechende Prozessalternative zu formulieren. Dies ließ sich mittels einer „Property“ des Typs „ReferenceProperty“ (vgl. hierzu Kapitel 4.2.2) erreichen. Weiterhin ist es beim Einlesen der Daten nötig, Prozessalternativen von anderen Arbeitsfolgen zu unterscheiden. Für diesen Fall kam eine „Property“ des Typs „SimpleProperty“ zur Differenzierung zum Einsatz. Einen Einblick in die Strukturen gewährt nachfolgende Abbildung 32.

```
<ProcessPlan>
  <Identifier>Prozessplan_Teil_1</Identifier>
  ...
  <Process>
    <Identifier>P3</Identifier>
    <Description>Löcher bohren</Description>
    ...
    <ResourcesRequired>
      <Resource>
        <ResourceIdentifier>Werk01_M3</ResourceIdentifier>
      </Resource>
    </ResourcesRequired>
    <OperationTime>
      <TimeUnit>second</TimeUnit>
      <Value>130.0</Value>
    </OperationTime>
    ...
    <Property>
      <Name>Ref_Prozessalternative</Name>
      <Description>Referenz zur Prozessalternative</Description>
      <ProcessReference>
        <ProcessPlanIdentifier>Prozessplan_Teil_1</ProcessPlanIdentifier>
        <ProcessIdentifier>P3_Alt1</ProcessIdentifier>
      </ProcessReference>
    </Property>
  </Process>
  <Process>
    <Identifier>P3_Alt1</Identifier>
    <Description>Löcher bohren</Description>
    ...
    <ResourcesRequired>
      <Resource>
        <ResourceIdentifier>Werk01_M4</ResourceIdentifier>
      </Resource>
    </ResourcesRequired>
    <OperationTime>
      <TimeUnit>second</TimeUnit>
      <Value>150.0</Value>
    </OperationTime>
    ...
    <Property>
      <Name>Ist_Prozessalternative</Name>
      <Description>Prozess ist eine Prozessalternative</Description>
      <Unit>Wahrheitswert</Unit>
      <Value>ja</Value>
    </Property>
  </Process>
</ProcessPlan>
```

Abbildung 32: Auszug des Prozessplanes aus der CMSD-Eingabedatei in XML

5.4.4 Resource Information Package

Mithilfe der Datenstrukturen des „Resource Information“ Paketes ließen sich Informationen zu Maschinen, Puffer sowie der Materialflussobjekte Quelle und Senke abbilden. Bei der Festlegung der Maschinendaten konnte berücksichtigt werden, dass die Umrüstzeiten vom aktuellen Rüstzustand der Maschine abhängen können. Zur Modellierung dieses Sachverhaltes kamen Rüstmatrizen zum Einsatz (vgl. Abbildung 33).



	string 0	time 1	time 2	time 3	time 4
string	TeileTyp	Teil_1	Teil_2	Teil_3	abgeruestet
1	Teil_1	0.0000	12.0000	15.0000	30.0000
2	Teil_2	11.0000	0.0000	17.0000	35.0000
3	Teil_3	13.0000	20.0000	0.0000	40.0000
4	abgeruestet	25.0000	35.0000	40.0000	0.0000

Abbildung 33: Beispiel einer Rüstmatrix des Anwendungsbeispiels

In der ersten Spalte der Rüstmatrix ist der aktuelle Konfigurationszustand der Maschine auszuwählen. Aus der Kopfzeile wird der gewünschte Rüstwechsel abgelesen. Ein Zeitwert in der Tabelle gibt an, wie lange der Umrüstvorgang von einem Rüstzustand zu einem anderen dauert.

Zur Modellierung der Rüstmatrix im CMSD-Format wurden die Entitätstypen „SetupDefinition“ und „SetupChangeoverDefinition“ eingesetzt (vgl. Anhang A.4). In Tabelle 3 ist anhand eines Beispiels beschrieben, wie die Entitätstypen genutzt werden können, um eine Rüstmatrix abzubilden. Ein ausführlicherer Auszug der CMSD-Eingabedatei zur Rüstmatrix, welcher die Beschreibung verdeutlicht, ist im Anhang C.2 zu finden.

Für weitere benötigte Merkmale der Maschinen konnte zum einen auf vordefinierte Strukturen zurückgegriffen werden, wie z. B. die Zuordnung der Maschinengruppenpuffer zu den Maschinen und der Verweis zum Schichtkalender. Zum anderen wurden „Properties“ vom Typ „SimpleProperty“ dazu verwendet, nicht vordefinierte Attribute anzulegen (Verfügbarkeit, MTTR, Maschinenzuverlässigkeit). Der Anhang C.1 bietet einen Überblick zur Beschreibung einer Maschine. Für die Hinterlegung der Bezeichnung von Strategien zur Reihenfolgegestaltung der Pufferobjekte wurde gleichfalls die „SimpleProperty“ eingesetzt.¹⁶

¹⁶ Vgl. hierzu die Abbildung dynamischer Zusammenhänge im CMSD-Format in Kapitel 4.3.2.

Auszug der Definitionen	Bemerkung
<pre> <Resource> <Name>M1</Name> ... <Property> <Name>Ruestmatrix</Name> <SetupDefinitionReference> <SetupDefinitionIdentifier>Setup_M1 </pre>	<ul style="list-style-type: none"> • Verweis (Referenz) auf die Rüstmatrix einer Maschine erfolgt mittels einer „Property“ innerhalb des Maschinenobjektes.
<pre> <SetupDefinition> <Identifier>Setup_M1</Identifier> <ChildSetup> <SetupDefinitionIdentifier>Setup_M1_Teil1 ... <ChildSetup> <SetupDefinitionIdentifier>Setup_M1_Teil2 ... </pre>	<ul style="list-style-type: none"> • Definition der Rüstmatrix (im Bsp. für Maschine 1) mithilfe des Objektes „SetupDefinition“. • Die Teiletypen 1 und 2 können auf Maschine 1 gerüstet werden.
<pre> <SetupDefinition> <Identifier>Setup_M1_Teil1</Identifier> <Property> <Value>Teil_1</Value> <Property> <SetupChangeoverIdentifier>Konfig_M1_Teil1 </pre>	<ul style="list-style-type: none"> • In weiteren Objekten der „SetupDefinition“ sind die Verweise zu den möglichen Rüstwechseln hinterlegt (Zeilen der Rüstmatrix). • Die obere „Property“ gibt an, für welchen Teiletyp die Maschine aktuell gerüstet ist.
<pre> <SetupChangeoverDefinition> <Identifier>Konfig_M1_Teil1</Identifier> <CurrentSetup> <SetupDefinitionIdentifier>Setup_M1_Teil1 <NewSetup> <SetupDefinitionIdentifier>Setup_M1_Teil2 <ChangeoverTime> <Value>15</Value> ... </pre>	<ul style="list-style-type: none"> • Rüstwechsel werden unter dem Objekt „SetupChangeover-Definition“ hinterlegt. • Unter „NewSetup“ wird im Bsp. die Zeit für den Wechsel von der Konfiguration für Teiletyp 1 auf 2 abgebildet.

Tabelle 3: Auszug der Definitionen für die Rüstmatrix der CMSD-Eingabedatei in XML

5.5 Fazit

Durch die prototypische Umsetzung eines automatisierten Modellgenerators konnten tiefergehende Einblicke zur Leistungsfähigkeit des CMSD Information Model gewonnen werden. Insbesondere wurde gezeigt, wie mithilfe der flexiblen Gestaltungsmöglichkeiten des CMSD-Standards einige grundlegende Sachverhalte bzw. Zusammenhänge von Fertigungssystemen abgebildet werden können. Der Entwurf von Arbeitsplänen, bei den Arbeitsgänge an mehreren Arbeitsstationen ausführbar sind (alternative Arbeitsfolgen), ließ sich u. a. veranschaulichen. Des Weiteren wurde die Nutzung des Layout Paketes zur Hinterlegung der Positionsdaten von Ressourcenobjekten demonstriert und darüber hinaus eine Formulierungsmöglichkeit zur Abbildung einer Rüstmatrix vorgestellt.

Eine Schwierigkeit bei der Nutzung des CMSD-Standards ist derzeit noch das aufwendige Formulieren eigener Ausgestaltungen der Datenstrukturen, um Zusammenhänge des fertigungstechnischen Umfeldes abzubilden. Den Initialaufwand würde eine Sammlung von Referenzstrukturen zu konkreten Anwendungen maßgeblich senken. Weiterhin ist es für den Anwender schwierig, ohne genauere Betrachtungen zu erkennen, wo die Grenzen der Leistungsfähigkeit des Standards liegen. Bei komplexeren Sachverhalten, wie z. B. das Abbilden des aktuellen Systemzustandes der Fertigung, wird das mühevoll Austesten kaum umgangen werden können.

Die zuletzt veröffentlichte Spezifikation des CMSD Information Model wurde noch nicht offiziell verabschiedet und gilt weiterhin als Entwurfsfassung, deshalb ist in der unmittelbaren Zukunft nicht mit einer Bereitstellung von Schnittstellen für die CMSD-Datenspezifikation durch die Hersteller betrieblicher Anwendungssysteme zu rechnen. Dies trägt dazu bei, dass der Standard noch eine Zeit lang nur unter hohem Initialaufwand genutzt werden kann. Für die Zukunft wird es erforderlich sein, eine einheitliche Verwendung der Datenstrukturen des Standards festzulegen und zu formalisieren, um die Implementierung homogener und günstiger Datenanbindungen an die betrieblichen Anwendungssysteme des Fertigungsumfeldes gewährleisten zu können. Anderenfalls werden die Datenanbindungen teure, individuelle Lösungen bleiben und eine größere Verbreitung des CMSD-Standards als Spezifikation zum Datenaustausch wird behindert.

Anhand eines überschaubaren Anwendungsbeispiels zur automatisierten Modellgenerierung ließ sich ein Eindruck davon gewinnen, dass der CMSD-Standard prinzipiell zur Abbildung simulationsrelevanter Informationen geeignet ist. Eine umfassende Einschätzung der Leistungsfähigkeit konnte jedoch infolge der notwendigen, starken Beschränkung des Betrachtungsfeldes nicht erfolgen.

Die prototypische Umsetzung soll allerdings dazu dienen, sich weitergehend mit dem CMSD-Standard auseinander zu setzen, um zu weiteren Erkenntnissen darüber zu gelangen, wie der Standard zu einem besseren Einbinden der Simulation in das Umfeld betrieblicher Anwendungssysteme beitragen kann. Mit dem Pilotszenario einer Werkstattfertigung wurde ein interessantes und anspruchsvolles Anwendungsgebiet für den CMSD-Standard aufgegriffen, welches zahlreiche Betätigungsfelder für nachfolgende Untersuchungen bietet.

Im anschließenden Kapitel wird auf zwei Erweiterungsmöglichkeiten für zukünftige Untersuchungen eingegangen, die im Zusammenhang mit dem Pilotszenario von Interesse sein könnten.

6 Vorschläge für Erweiterungen

In diesem Kapitel sollen Anregungen für weitergehende Betrachtungen vorgeschlagen werden, die lohnenswert erscheinen und zur Fortentwicklung des Prototypen beitragen.

6.1 Abbildung von Strategien durch „Microfunktionen“

Bei der Anwendung von Strategien der Produktionssteuerung werden in der Praxis selten Regeln in ihrer Grundform verwendet [Se05, S.87f]. Der Grund hierfür ist, dass die Grundregeln unterschiedlich gut zur Erreichung einzelner Ziele (z. B. hohe Auslastung, kurze Durchlaufzeiten) geeignet sind. Sie werden deshalb oft in Kombination eingesetzt [Ne07, S.699]. Um das reale Fertigungssystem wirklichkeitsgetreuer im Simulationsmodell abbilden zu können, erscheint es naheliegend diesen Aspekt zu berücksichtigen.

Selke [Se05] greift deswegen den Vorschlag der „Micro-functions“ von Wiedemann [Wi99] auf und lässt ihn in seine Arbeit einfließen. Die Grundidee soll auch in diesen Erweiterungsvorschlag übernommen werden. Der Gedanke hierbei ist, eine komplexere Strategie aus ihren Grundbestandteilen (Grundregeln) aufzubauen und als Komposition der einzelnen Bestandteile im Simulationsmodell einzusetzen.

Auf den Prototypen übertragen, könnte untersucht werden, wie eine Bibliothek von Grundregeln zu erstellen und im Simulationsmodell einzubinden wäre, damit die zuvor genannte Idee verwirklicht werden kann. Das Ziel ist, zur Formulierung einer Strategie auf einen Pool von Grundregeln zurückgreifen zu können, welche parametrisierbar sind, um größtmögliche Flexibilität zu erreichen.

Um diesen Vorschlag anhand des Prototypen umsetzen zu können, werden einige Änderungen notwendig sein. Damit eine bessere Flexibilität bei der Nutzung von Strategien der Produktionssteuerung erreichbar ist, sollte auf die Verwendung des Standardbausteins „Sortierer“ als Puffer verzichtet werden. Dieser erwies sich als recht unflexibel, was die Implementierung eigener Sortierkriterien betrifft. Als besser geeignet erscheint es, den Puffer mithilfe einer eigenen Tabellenstruktur zu verwalten und diese für regelbasierte Sortiervorgänge zu nutzen. Diese Verfahrensweise wird z. B. in der von Siemens PLM Software vertriebenen Bibliothek „Shop“ angewendet, welche zur Modellierung werkstattorientierter Fertigungen eingesetzt werden kann.

6.2 Online-Kopplung zur Produktionssteuerung

Eine andere Erweiterung betrifft die in Kapitel 5.1 angesprochene Online-Kopplung zur Produktionssteuerung beim betriebsbegleitenden Einsatz (vgl. Abbildung 34). Ziel hierbei ist, den aktuellen Zustand des Fertigungssystems zur Simulationsanwendung zu übertragen, um zu erreichen, dass das Simulationsmodell mit der Initialisierung in einen Zustand versetzt wird, der dem des realen Fertigungssystems möglichst nahe kommt. Dadurch soll die Simulation in die Lage versetzt werden, zukünftiges Geschehen des realen Systems genauer vorwegzunehmen. Bei der Betrachtung im akademischen Rahmen wäre es möglich, auf BDE-Daten einer emulierten Fertigung als Datengrundlage zurückzugreifen.

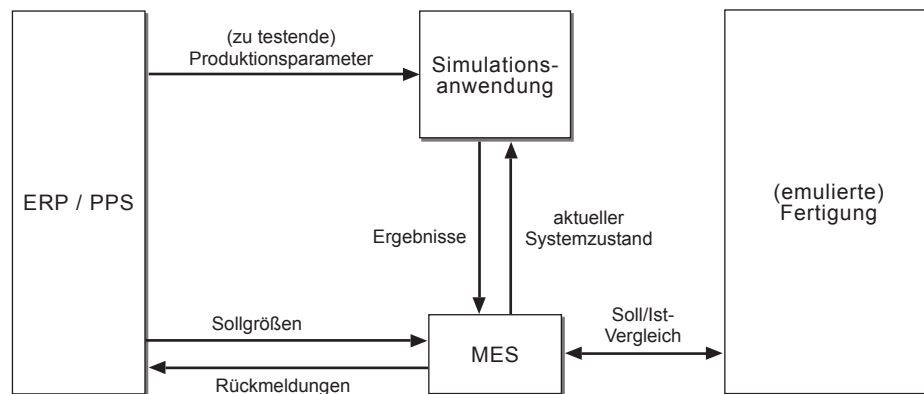


Abbildung 34: Online-Anbindung der Simulation an die Produktionssteuerung

Dabei stellt sich bei der Verwendung des CMSD-Standards in diesem Zusammenhang eine wichtige Frage: Lässt sich der aktuelle Zustand des Fertigungssystems ausreichend mittels der Datenstrukturen des Standards abbilden? Zu beachtende Aspekte sind z. B. bereits begonnene Aufträge, deren Restbearbeitungszeit zu hinterlegen ist, derzeitige Maschinenstörungen, begonnene Rüstvorgänge oder Wartungsmaßnahmen.

Ein weiterer Sachverhalt, welcher die Beantwortung der obigen Frage erschwert, ist die Anwendung von Strategien der Produktionssteuerung in der realen Fertigung. Denn durch deren Einsatz kann das Fertigungssystem in aufwendig zu beschreibende Zustände versetzt werden. Als Beispiel hierzu kann das Strategiecluster der Losgrößengestaltung dienen (vgl. Tabelle 1 in Kapitel 2.3.5). Schwierig abzubildende Systemzustände können z. B. durch das Splitten oder Zusammenfassen von Auftragslosen entstehen, wenn diese zu einem späteren Zeitpunkt wieder in ihre Ausgangskonstellation überführt werden sollen.

Die Herausforderung der Online-Kopplung besteht darin, dem Simulationssystem möglichst viele, für die Abbildung des aktuellen Zustandes relevante, Informationen zur Verfügung zu stellen.

7 Zusammenfassung

Der theoretische Teil der Arbeit hat zum Ziel, eine Einschätzung bezüglich des „Core Manufacturing Simulation Data“ (CMSD) Information Model als Spezifikation für ein Datenaustauschformat und dessen grundsätzliche Eignung für die automatisierte Erstellung von Simulationsmodellen, abzugeben. Eine der Aufgaben zur Realisierung dieses Zieles ist, die Untersuchung des CMSD Information Model hinsichtlich seiner Möglichkeiten zur Abbildung simulationsrelevanter Eingangsdaten sowie die Identifizierung der Stärken und Schwächen des Standards, um Aussagen bezüglich der Nutzbarkeit, insbesondere im Hinblick der Nutzung zur automatisierten Simulationsmodellgenerierung, treffen zu können.

Im praktischen Teil wird das Ziel der Vertiefung zuvor getätigter theoretischer Betrachtungen und die Erstellung einer prototypischen Umsetzung eines automatisierten Modellgenerators verfolgt. Dazu wird ein Prototyp einer automatisierten Modellgenerierung entworfen. Das zweite Kapitel dient zur Einführung in den Themenkomplex der vorliegenden Arbeit. In diesem werden zunächst für den Betrachtungsgegenstand wichtige Begriffe eingeführt und erläutert. Nach der Festlegung eines Begriffsverständnisses zu den Bezeichnungen Fertigungsplanung sowie Produktionsplanung und -steuerung, findet eine Betrachtung der Ablaufsimulation im fertigungstechnischen Umfeld statt. Hierbei rücken speziell die Anwendungsgebiete und die relevanten Eingangsdaten der Ablaufsimulation in den Blickpunkt der Betrachtungen. Im Anschluss erfolgt die Begriffsklärung der Digitalen Fabrik und ein Überblick zu dieser.

Im dritten Kapitel wird der Hintergrund für die automatisierte Erstellung von Simulationsmodellen beleuchtet. Hierzu werden die Anwendungsfelder zum planungs- und betriebsbegleitenden Einsatz beschrieben. Zudem können mit der Betrachtung zweier aktueller Ansätze aus der Literatur die Schwachpunkte derzeitiger Ansätze zur automatisierten Modellgenerierung herausgestellt werden. Es zeigt sich, dass der hohe Aufwand zum Datenaustausch zwischen betrieblichen Anwendungssystemen und der Simulation ein maßgebliches Hemmnis für die Anwendung darstellt.

Im darauffolgenden vierten Kapitel wird ein recht neues Informationsmodell für den Datenaustausch basierend auf dem XML-Format vorgestellt (CMSD Information Model). Dieses zielt darauf ab, eine gegenüber der heutigen Situation bessere Integration der Simulation in das Umfeld betrieblicher Anwendungssysteme zu ermöglichen. Im Blickpunkt der Betrachtung steht die Frage, ob das CMSD Information Model einen positiven Beitrag zur Datenübertragung automatisierter Modellgeneratoren leisten kann. Untersuchungsgegenstand sind dabei die Möglichkeiten der Abbildung statischer und dynamischer Zusammenhänge von Fertigungssystemen.

Das vorletzte Kapitel ist der Beschreibung der implementierten prototypischen Umsetzung eines automatisierten Modellgenerators gewidmet. Ein erarbeitetes Szenario, das als Grundlage für den Prototyp dient, verdeutlicht, wie eine integrierte Systemlösung prinzipiell ausse-

hen könnte. Zur Demonstration des Prototypen wird eine Beispielanwendung formuliert und umgesetzt. Die für die Anwendung benötigten Eingangsdaten werden in konkrete Beschreibungen der CMSD-Datenstrukturen überführt und im Rahmen dieses Kapitels vorgestellt. Das abschließende sechste Kapitel beinhaltet zwei Erweiterungsvorschläge für den Prototypen, die als lohnenswert eingeschätzt werden. Zum einen handelt es sich um die Abbildung von komplexeren Strategien mittels der Komposition einzelner Grundbausteine von Strategien, sogenannte „Microfunktionen“. Zum anderen ist von einer Online-Kopplung zur Produktionsteuerung die Rede, um das Simulationsmodell zu Beginn von Simulationsexperimenten in einen Zustand zu überführen, der dem des realen Fertigungssystems möglichst nahe kommt. Dies soll den Simulator im Rahmen des betriebsbegleitenden Einsatzes in die Lage versetzen, zukünftiges Geschehen genauer vorherzusagen.

Literaturverzeichnis

- [BäHa03] *Bär, Thomas; Haasis, Siegmund*: Prozessplanung, Prozessmodellierung und -simulation - ein Überblick. In: *Bayer, Johann; Collisi, Thomas; Wenzel, Sigrid (Hrsg.)*: Simulation in der Automobilproduktion. Springer Verlag, Berlin 2003; S. 151-163.
- [BFW00] *Bley, Helmut; Franke, Christina; Wuttke, Claas Christian*: Integration der operativen Ebenen in Simulationsprojekte. In: Industrie Management, Band 16, Heft 2, Gito Verlag, Berlin 2000; S. 13-17.
- [BFZ06] *Bley, Helmut; Fritz, Jürgen; Zenner, Christian*: Die zwei Seiten der Digitalen Fabrik - Software-Werkzeug und Methode. In: Zeitschrift für wirtschaftlichen Fabrikbetrieb, Jahrg. 101, Carl Hanser Verlag, München 2006; S. 19-23.
- [BRK04] *Bierschenk, Sabine; Ritter, Arno; Kuhlmann, Timm*: Status quo - Digitale Fabrik - auch für kleine und mittlere Unternehmen? Aktuelle Studie des Fraunhofer IPA. In: Digital Engineering, Ausgabe 5/2004, WIN-Verlag, München 2004; S. 26-28.
- [Da01] *Dangelmaier, Wilhelm*: Fertigungsplanung: Planung von Aufbau und Ablauf der Fertigung; Grundlagen, Algorithmen und Beispiele, 2. Auflage. Springer-Verlage, Berlin 2001.
- [Da09] *Dangelmaier, Wilhelm*: Theorie der Produktionsplanung und -steuerung - Im Sommer keine Kirschpralinen? Springer-Verlag, Berlin 2009.
- [Ec02] *Eckardt, Frank*: Ein Beitrag zu Theorie und Praxis datengetriebener Modellgeneratoren zur Simulation von Produktionssystemen. Shaker Verlag, Aachen 2002.
- [Ev96] *Eversheim, Walter*: Organisation in der Produktionstechnik, Band 1 - Grundlagen, 3. Auflage. VDI-Verlag, Düsseldorf 1996.
- [Ev02] *Eversheim, Walter*: Organisation in der Produktionstechnik 3 - Arbeitsvorbereitung, 4. Auflage. Springer-Verlag, Berlin 2002.

- [Fr03] *Franke, Christina*: Feature-basierte Prozesskettenplanung in der Montage als Basis für die Integration von Simulationswerkzeugen in der Digitalen Fabrik. Schriftenreihe Produktionstechnik Band 28, Universität des Saarlandes, Saarbrücken 2003.
- [Fr07] *Fritz, Jürgen Ulrich*: Zielführende Modellierung und Analyse schlanker Fertigungssysteme mit der Digitalen Fabrik. Schriftenreihe Produktionstechnik Band 41, Universität des Saarlandes, Saarbrücken 2007.
- [Go89] *Gottschalk, Eberhard*: Rechnergestützte Produktionsplanung und -steuerung. Verlag Technik, Berlin 1998.
- [HeMi03] *Heisel, Uwe; Michaelis, Michael*: Rekonfiguration von Produktionssystemen. In: *Bullinger, Hans-Jörg; Warnecke, Hans Jürgen; Westkämper Engelbert (Hrsg.)*: Neue Organisationsformen im Unternehmen - Ein Handbuch für das moderne Management, 2. Auflage, Springer-Verlag, Berlin 2003; S. 526–542.
- [HrAc98] *Hrdliczka, Veronika; Acél, Peter*: Simulieren geht über Probieren - aber nicht ohne studieren! Betriebswissenschaft und Innovation, ETH-Zentrum für Unternehmenswissenschaften (BWI), Management Verlag, Zürich 1998.
- [Je07] *Jensen, Sven*: Eine Methodik zur teilautomatisierten Generierung von Simulationsmodellen aus Produktionsdatensystemen am Beispiel einer Job Shop Fertigung. Unidruckerei der Universität Kassel, Kassel 2007.
- [Jo07] *Johansson, Marcus et. al.*: A Test Implementation of the Core Manufacturing Simulation Data Specification. In: Proceedings of the 2007 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 2007; S. 1673-1681.
- [Ku05] *Kurbel, Karl*: Produktionsplanung und -steuerung im Enterprise Resource Planning and Supply Chain Management, 6. Auflage. Oldenbourg Verlag, München 2005.
- [Kü06] *Kühn, Wolfgang*: Digitale Fabrik Fabriksimulation für Produktionsplaner. Carl Hanser Verlag, München 2006.

- [LLR06] *Leong, Swee; Lee, Y. Tina; Riddick Frank*: A Core Manufacturing Simulation Data Information Model for Manufacturing Applications. In: Proceedings of the Systems Interoperability Standards Organization 2006 - Fall Simulation Interoperability Workshop, Manufacturing Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, 2006.
- [Ma05] *Marczinski, Götz*: Digitale Fabrik - mit dem 4-Stufenmodell zum Erfolg. In: PPS Management, Heft 2/2005, Gito Verlag, Berlin 2005; S. 38-41.
- [Mö08] *Mönch, Lars*: Betriebsdatenerfassung In: Enzyklopädie der Wirtschaftsinformatik - Online-Lexikon, Abruf am 27.08.2010 um 12:30 Uhr, <http://www.oldenbourg.de:8080/wi-enzyklopaedie/lexikon/informationssysteme/Sektor-spezifische-Anwendungssysteme/Produktionsplanungs--und--steuerungssystem/Manufacturing-Execution-System/Betriebsdatenerfassung>
- [Ne07] *Nebi, Theodor*: Produktionswirtschaft, 6. Auflage. Oldenbourg Verlag, München 2007.
- [Ra03] *Rabe, Markus*: Modellierung von Layout und Steuerungsregeln für die Materialfluss-Simulation. Fraunhofer IRB Verlag, Stuttgart 2003.
- [ReFe97a] *Reinhart, Gunther; Feldmann, Klaus*: Simulation - Schlüsseltechnologie der Zukunft? Stand und Perspektiven, Herbert Utz Verlag, München 1997.
- [ReFe97b] *Reinhart, Gunther; Feldmann, Klaus (Hrsg.)*: Simulationsbasierte Entscheidungshilfsmittel für Organisation und Produktion - Ergebnisbericht zum ersten Forschungsjahr des Forschungsverbunds Simulationstechnik (ForSIM), Hochschul-Eigenverlag, Erlangen 1997.
- [RiLe08] *Riddick, Frank; Lee, Y. Tina*: Representing Layout Information in the CMSD Specification. In: Proceedings of the 2008 Winter Simulation Conference, Manufacturing Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, 2008; S. 1777-1784.
- [Ro09] *Rooks, Tobias*: Rechnergestützte Simulationsmodellgenerierung zur dynamischen Absicherung der Montagelogistikplanung bei der Fahrzeugneutypplanung im Rahmen der Digitalen Fabrik. Shaker Verlag, Aachen 2009.

- [Se05] *Selke, Carsten*: Entwicklung von Methoden zur automatischen Simulationsmodellgenerierung. Forschungsberichte iwv Band 193, Herbert Utz Verlag, München 2005.
- [SISO10] *Leong, Swee et. al. - Core Manufacturing Simulation Data Product Development Group*: Standard for: Core Manufacturing Simulation Data - UML Model. Simulation Interoperability Standards Organization (SISO), Inc., Orlando, Florida, 2010.
- [Sp95] *Splanemann, Ralph*: Teilautomatische Generierung von Simulationsmodellen aus systemneutral definierten Unternehmensdaten am Beispiel der Integration der Materialflußsimulation in die Planung von Fertigungsanlagen. Wissenschaftsverlag, Aachen 1995.
- [VDI 3633-1] *Verein Deutscher Ingenieure (Hrsg.)*: VDI-Richtlinie 3633, Blatt 1: Simulation von Logistik-, Materialfluß- und Produktionssystemen - Grundlagen. Entwurf der überarbeiteten Fassung. VDI-Handbuch Materialfluß und Fördertechnik, Band 8. Beuth Verlag, Berlin 2000.
- [VDI 3633-5] *Verein Deutscher Ingenieure (Hrsg.)*: VDI-Richtlinie 3633, Blatt 5: Simulation von Logistik-, Materialfluß- und Produktionssystemen - Integration der Simulation in die betrieblichen Abläufe. VDI-Handbuch Materialfluß und Fördertechnik, Band 8. Beuth Verlag, Berlin 2000.
- [VDI 4499] *Verein Deutscher Ingenieure (Hrsg.)*: VDI-Richtlinie 4499, Blatt 1: Digitale Fabrik - Grundlagen. VDI-Handbuch Materialfluss und Fördertechnik, Band 8. Beuth Verlag, Berlin 2008.
- [VDI 5600] *Verein Deutscher Ingenieure (Hrsg.)*: VDI-Richtlinie 5600, Blatt 1: Fertigungsmanagementsysteme - Manufacturing Execution Systems (MES). VDI-Handbuch Informationstechnik, Beuth Verlag, Berlin 2007.
- [Wi99] *Wiedemann, Thomas*: Database Oriented Modeling with Simulation Microfunctions. In: *Farrington, P. A. et. al. (Hrsg.)*: Proceedings of the 1999 Winter Simulation Conference, IEEE, 1999, S. 586 – 590.
- [Wi05] *Wiendahl, Hans-Peter*: Betriebsorganisation für Ingenieure, 5. Auflage. Carl Hanser Verlag, München 2005.

- [WaWe06] *Warnecke, Hans-Jürgen; Westkämpfer, Engelbert*: Einführung in die Fertigungstechnik, 7. Auflage. Teubner Verlag, Wiesbaden 2006.
- [Wu00] *Wuttke, Class Christian*: Mehrfachnutzung von Simulationsmodellen in der Produktionslogistik. Schriftenreihe Produktionstechnik Bd. 20, Universität des Saarlandes, Saarbrücken 2000.
- [Ze06] *Zenner, Christian*: Durchgängiges Variantenmanagement in der Technischen Produktionsplanung. Schriftenreihe Produktionstechnik Bd. 37, Universität des Saarlandes, Saarbrücken 2006.

Abbildungsverzeichnis

Abbildung 1: Aufgaben der Fertigungsplanung (Arbeitsplanung) nach [Ev96, S.74]	4
Abbildung 2: PPS als Baustein innerhalb der Fertigung nach [Ze06, S.15].....	6
Abbildung 3: Funktionen der Produktionsplanung und -steuerung [Ze06, S.16].....	6
Abbildung 4: Einsatz der Simulation zur Unterstützung der operativen Produktionsplanung [Kü06, S.90].....	11
Abbildung 5: Anwendungsgebiete der Ablaufsimulation nach [ReFe97a, S.19]	11
Abbildung 6: Phasen einer Simulationsstudie nach [VDI3633-1, S.10f; Ro09, S.48]	13
Abbildung 7: Gruppierung der Eingangsdaten für Ablaufsimulationen nach [VDI 3633-1, S.12f; Se05, S.15f]	15
Abbildung 8: Möglichkeiten der Nutzung stochastischer Daten für die Simulation [Kü06, S.153]	17
Abbildung 9: Fokus der Digitalen Fabrik nach [VDI 4499, S.3].....	18
Abbildung 10: Struktur der Digitalen Fabrik nach [BRK04, S.28] aus [Ro09, S.7]	19
Abbildung 11: Anwendungsfelder für Werkzeuge der Digitalen Fabrik nach [BäHa03, S.152]	20
Abbildung 12: Vorgehensmodell zur Interpretation von Strategien und Abläufen der Produktionssteuerung [Se05, S.95]	28
Abbildung 13: Vorgehensmodell des Ansatzes von Jensen [Je07, S.5]	29
Abbildung 14: CMSD als neutrales Datenformat zwischen betrieblichen Anwendungssystemen [Jo07, S.1674]	33
Abbildung 15: Pakete des CMSD-Models nach [SISO10, S.18].....	34
Abbildung 16: Beispiele für Grundtypen des CMSD Information Model [SISO10, S.54 u.57]	35
Abbildung 17: Ausschnitt der Basic Entitätsreferenzklassen [SISO10, S.79].....	36
Abbildung 18: Ausschnitt der Complex Entitätsreferenzklassen [SISO10, S.81]	36
Abbildung 19: Property Klasse des Basic Structure Package [SISO10, S.21]	37
Abbildung 20: Klassendefinitionen der Layoutelemente [SISO10, S.88]	38
Abbildung 21: Klassendefinitionen der Formbeschreibungen [SISO10, S.89]	39
Abbildung 22: Ausschnitt der Klassendefinition zum Prozessplan [SISO10, S.133].....	41
Abbildung 23: Klassendefinition zu Ressourcen und -klassen [SISO10, S.144]	42
Abbildung 24: Integration der Simulationsanwendung in das betriebliche Umfeld.....	47
Abbildung 25: Gesamtkonzept des Modellgenerators in Anlehnung an [Ro09, S.147].....	48
Abbildung 26: Übersicht der Bestandteile des Modellgenerators	49
Abbildung 27: Datentabelle der Maschinendaten.....	51
Abbildung 28: Anwendungsbeispiel für den Modellgenerator.....	53
Abbildung 29: Grundaufbau eines CMSD-Dokuments [SISO10, 19].....	54

Abbildung 30: Auszug der Layoutinformationen aus der CMSD-Eingabedatei in XML ..	55
Abbildung 31: Darstellung eines Auftrages aus der CMSD-Eingabedatei in XML	55
Abbildung 32: Auszug des Prozessplanes aus der CMSD-Eingabedatei in XML.....	56
Abbildung 33: Beispiel einer Rüstmatrix des Anwendungsbeispiels	57
Abbildung 34: Online-Anbindung der Simulation an die Produktionssteuerung.....	61
Abbildung 35: Basic Entitätsreferenzklassen [SISO10, S.79].....	72
Abbildung 37: Klassendefinitionen zur Abbildung von Restriktionen zu Anordnungsbeziehungen [SISO10, S.23]	73
Abbildung 36: Complex Entitätsreferenzklassen [SISO10, S.81]	73
Abbildung 38: Klassendefinitionen zu Kundenaufträgen und betriebsinternen Aufträgen [SISO10, S.120]	74
Abbildung 39: Klassendefinitionen zum Ablaufplan [SISO10, S.122]	75
Abbildung 40: Klassendefinitionen zum Schichtkalender [SISO10, S.132]	75
Abbildung 42: Klassendefinitionen für Ressourcengruppen und Mitarbeiterfähigkeiten [SISO10, S.145]	76
Abbildung 41: Klassendefinitionen zum Prozess- und Wartungsplan [SISO10, S.133]	76
Abbildung 43: Klassendefinitionen für Konfigurationszustände der Ressourcen [SISO10, S.146]	77

Tabellenverzeichnis

Tabelle 1: Einteilung der Strategien und Abläufe der Produktionssteuerung nach [Se05, S.45ff]	16
Tabelle 2: Anwendung der Simulationstechnik in verschiedenen Bereichen produzierender Unternehmen nach [Kü06, S.56]	21
Tabelle 3: Auszug der Definitionen für die Rüstmatrix der CMSD-Eingabedatei in XML.	58

Anhang

A Ausgewählte Klassendefinitionen des CMSD Information Model - UML-Teil

A.1 Support Package

Entity Reference Definition Package

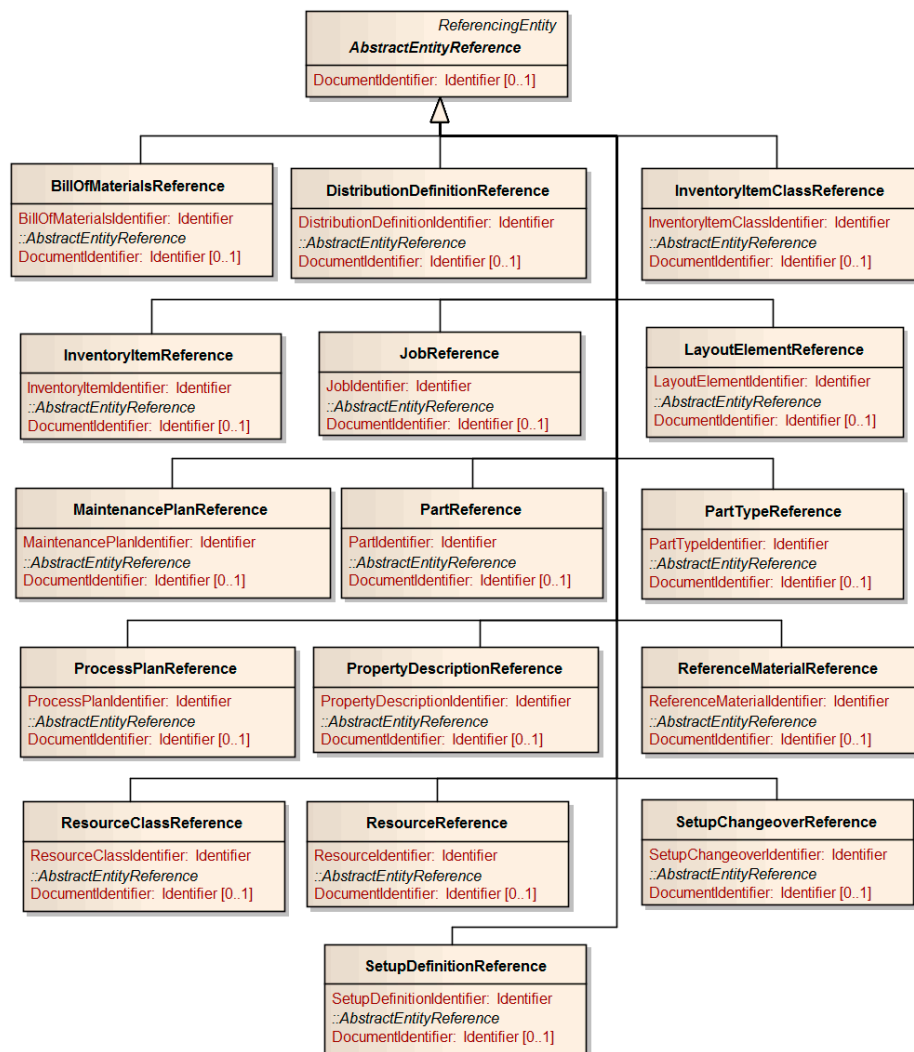


Abbildung 35: Basic Entitätsreferenzklassen [SISO10, S.79]

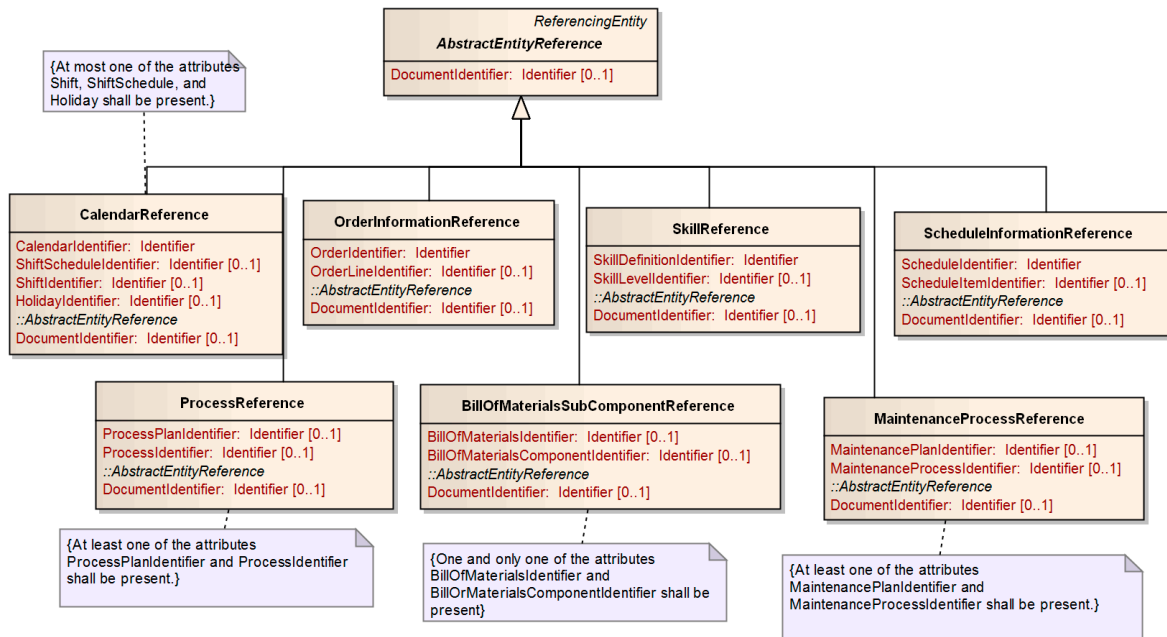


Abbildung 36: Complex Entitätsreferenzklassen [SISO10, S.81]

Basic Structures Package

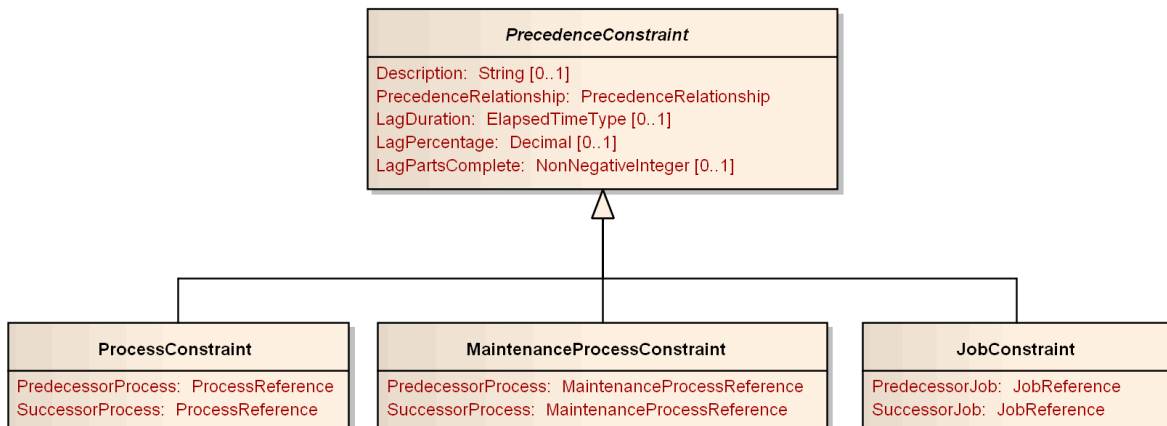


Abbildung 37: Klassendefinitionen zur Abbildung von Restriktionen zu Anordnungsbeziehungen [SISO10, S.23]

A.2 Production Operations Package

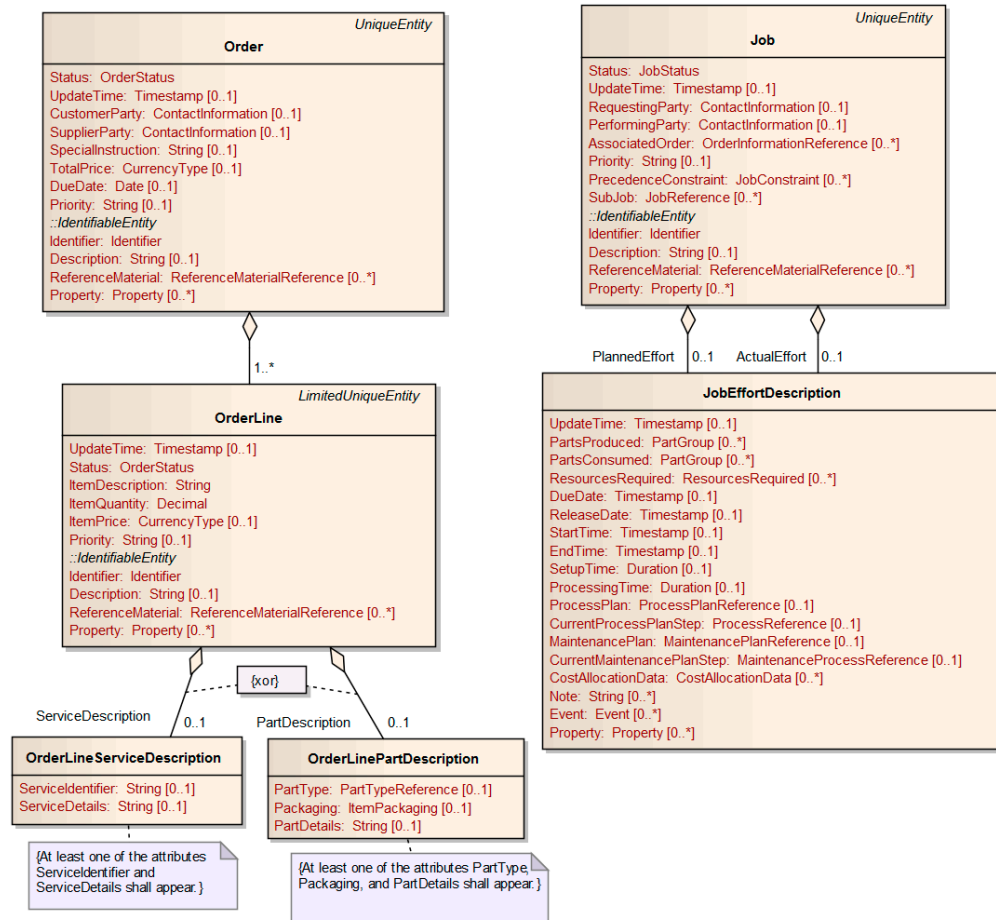


Abbildung 38: Klassendefinitionen zu Kundenaufträgen und betriebsinternen Aufträgen [SISO10, S.120]

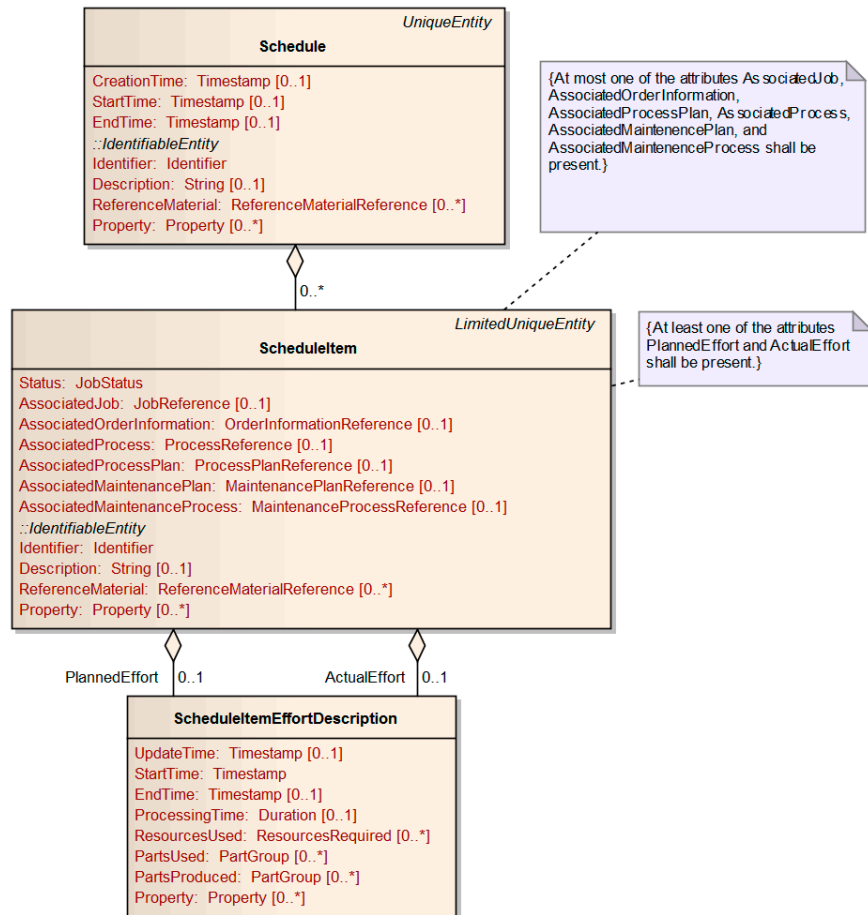


Abbildung 39: Klassendefinitionen zum Ablaufplan [SISO10, S.122]

A.3 Production Planning Package

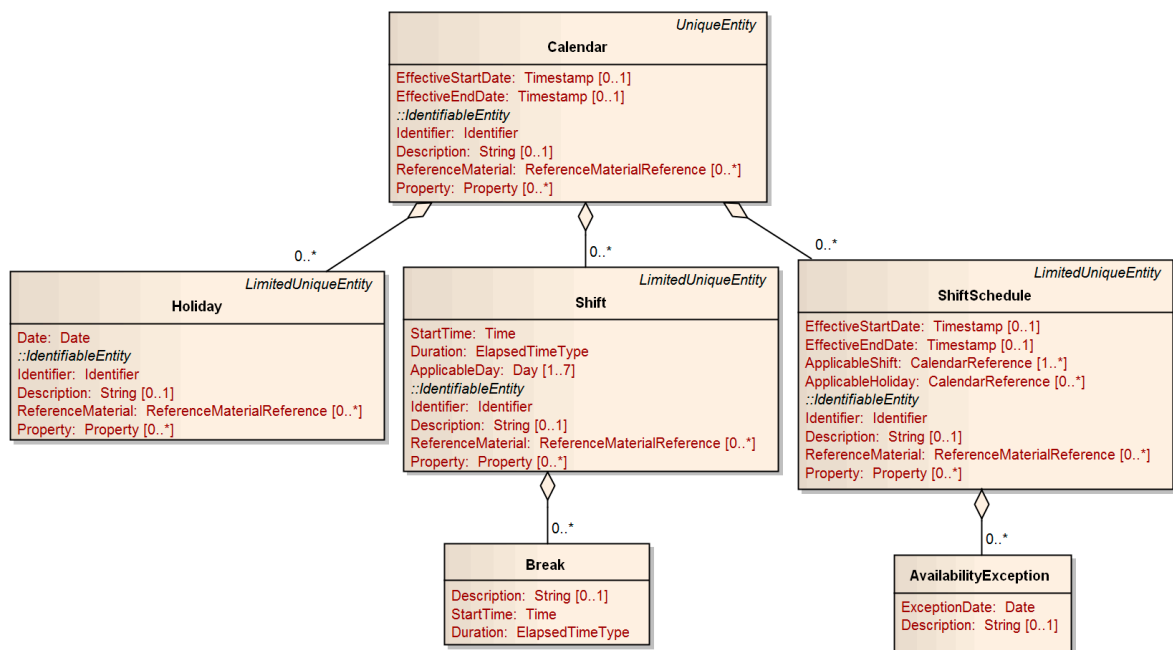


Abbildung 40: Klassendefinitionen zum Schichtkalender [SISO10, S.132]

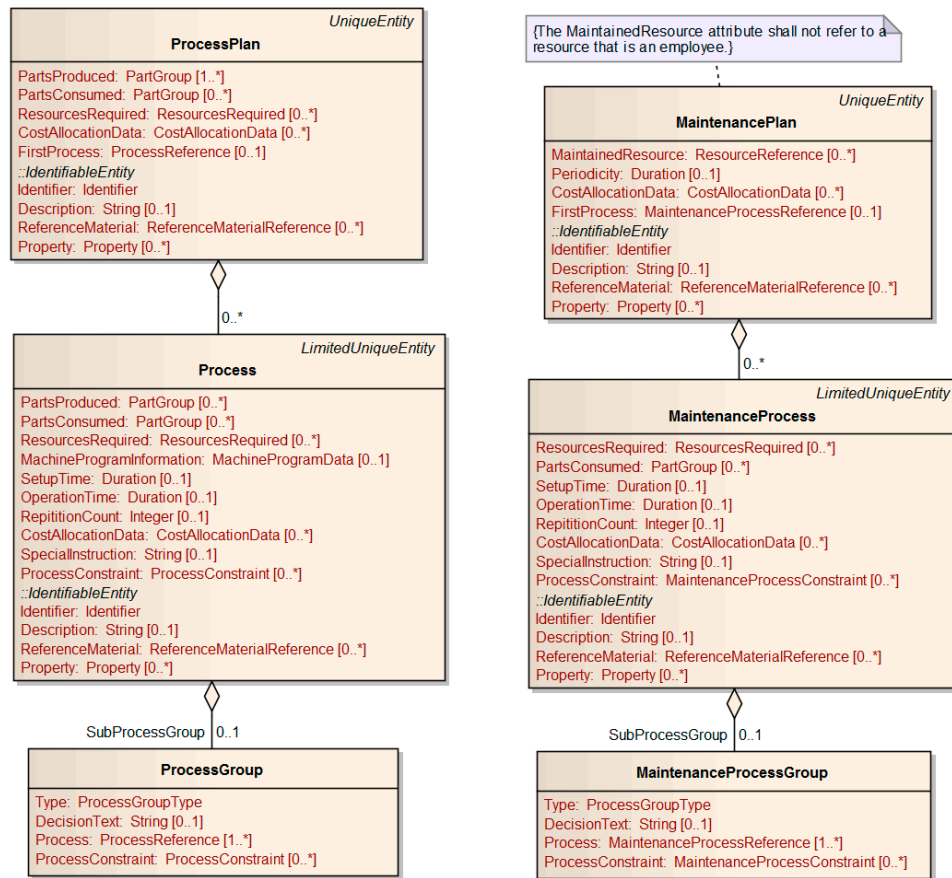


Abbildung 41: Klassendefinitionen zum Prozess- und Wartungsplan [SISO10, S.133]

A.4 Resource Information Package

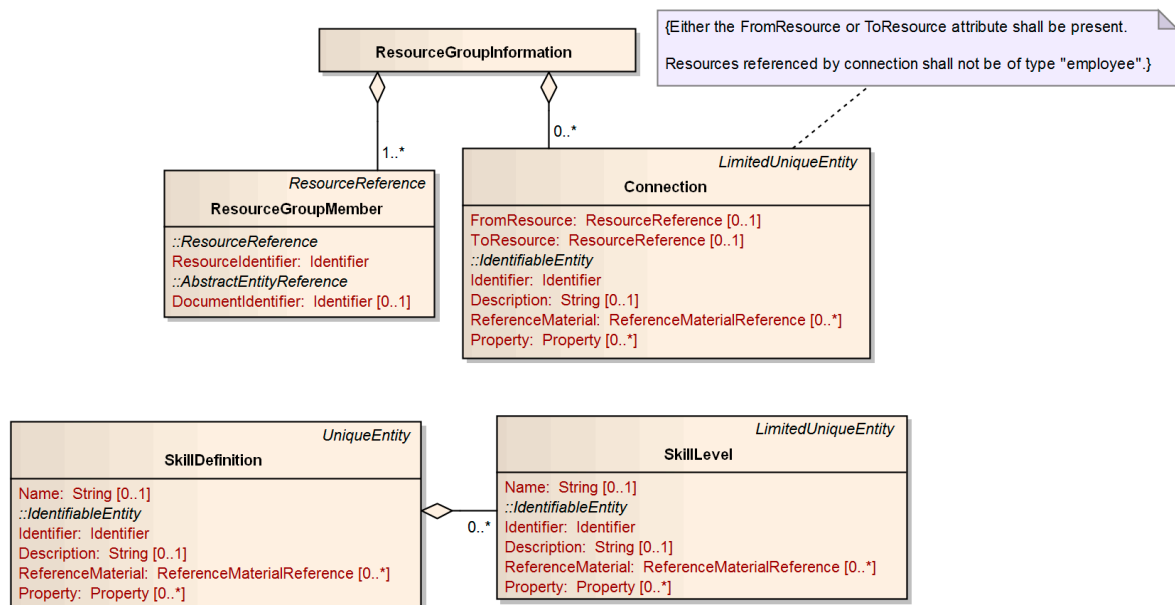


Abbildung 42: Klassendefinitionen für Ressourcengruppen und Mitarbeiterfähigkeiten [SISO10, S.145]

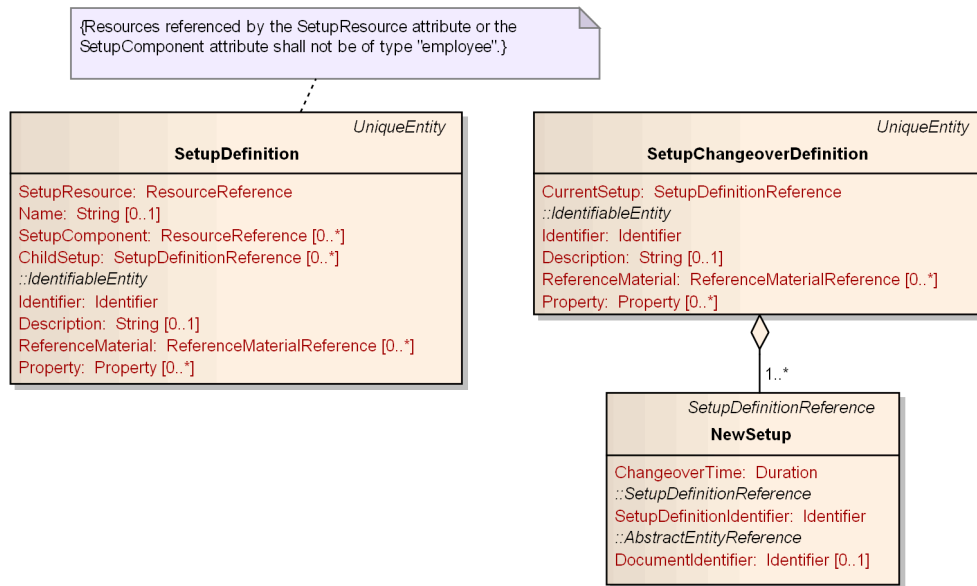


Abbildung 43: Klassendefinitionen für Konfigurationszustände der Ressourcen [SISO10, S.146]

B Ausgewählte Importmethoden des Prototypen

B.1 Methode zum Einlesen des Arbeitsplanes

```

/* Importmethode des Modellgenerators
Daten werden durch eine CMSD-XML-Datei (Spezifikation Stand 30.04.2010) bereitgestellt.
Einlesen des Arbeitsplanes in die Verwaltungstabelle Arbeitsplan */
-----

(Pfad_XML_Datei:string)

is
  tabName      : object;
  i,j,k,l      : integer;
  tmpStr,tmpRes : string;
do
  tabName := Arbeitsplan;
  tabName.loeschen({0,1}..{tabName.xdim,*});
  XMLSchnittstelle.öffneDokument;
  XMLSchnittstelle.selektiereKnoten („/CMSDDocument/DataSection/ProcessPlan“);
  i := 0;

  while XMLSchnittstelle.holeNächstenKnoten loop
    XMLSchnittstelle.selektiereKinder;
    i := i+1;
    j := 0;

    While XMLSchnittstelle.holeNächstenKnoten loop

      /* Einlesen der Teiletypen und Initialisierung der Arbeitsschrittetabelle */
      -----
      XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/ProcessPlan[„,i,“]/PartsProduced/PartType/
      PartTypeIdentifier“));
      XMLSchnittstelle.holeNächstenKnoten;

      tabName[„Teile Typ“,i] := XMLSchnittstelle.holeKnotenWert;
      tabName.erzeugeSubListe(1,i);
      tabName[„Arbeitsschritte“,i].setName(to_str („Arbeitsschritte für „,tabName[„Teile Typ“,i]));

      /* Einlesen der Prozessdaten in Arbeitsschrittetabelle */
      -----
      XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/ProcessPlan[„,i,“]/Process“));

      while XMLSchnittstelle.holeNächstenKnoten loop
        XMLSchnittstelle.selektiereKinder;
        k := tabName[„Arbeitsschritte“,i].ydim+1;
        j := j+1;

        /* Überprüfung, ob der aktuelle Prozess eine Prozessalternative ist */
        -----
        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/ProcessPlan[„,i,“]/Process[„,j,“]/
        Property[contains(\"Ist Prozessalternative\",Name)]/Child::Value“));
        XMLSchnittstelle.holeNächstenKnoten;
        tmpStr := XMLSchnittstelle.holeKnotenWert;

        /* Aktuelle Prozess ist keine Prozessalternative. Einlesen der ID und der Arbeitsschrittbezeichnung */
        -----

```

```

        If tmpStr = „" or tmpStr = „0" or toLower(tmpStr) /= „ja" and toLower(tmpStr) /= „1" then

XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/ProcessPlan[„,i,“]/Process[„,j,“]/
Identifier));
XMLSchnittstelle.holeNächstenKnoten;
tabName[„Arbeitsschritte“,i][„ID“,k] := XMLSchnittstelle.holeKnotenWert;

XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/ProcessPlan[„,i,“]/Process[„,j,“]/
Description));
XMLSchnittstelle.holeNächstenKnoten;
tabName[„Arbeitsschritte“,i][„Arbeitsschritt“,k] := XMLSchnittstelle.holeKnotenWert;

/* Einlesen der Einschleusestation: Jede Maschine hat als Vorgänger einen Puffer, entweder direkt
oder indirekt über eine Flussssteuerung. Dieser vorgeschaltete Puffer ist die Einschleusestation.
Bestimmung des Vorgängers (Koten: „FromResource“) */
-----

If tabName[„Arbeitsschritte“,i][„Einschleusestation“,k] = void then

    tmpRes := tabName[„Arbeitsschritte“,i][„ID“,k];
    XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/ProcessPlan[„,i,“]/
Process[contains („,to_str („,“,“,“, Identifier)]/ResourcesRequired/Resource/ResourceIdentifier));
    XMLSchnittstelle.holeNächstenKnoten;
    tmpRes := XMLSchnittstelle.holeKnotenWert;

    XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Resource[contains („,to_str („,“,“,“, Identifier)]/GroupDefinition/Connection/FromResource/ResourceIdentifier));
    XMLSchnittstelle.holeNächstenKnoten;
    tabName[„Arbeitsschritte“,i][„Einschleusestation“,k] := Hilfsmethoden.Liefere_Name („Resource“,XMLS
chnittstelle.holeKnotenWert,true);
    end;

    If tabName[„Arbeitsschritte“,i][„mögliche Stationen“,k] = void then
        tabName[„Arbeitsschritte“,i].erzeugeSubListe(2,k);
        tabName[„Arbeitsschritte“,i][„mögliche Stationen“,k].setzeName („Stationen");
    end;

end;

/* Einlesen der möglichen Bearbeitungsstationen */
-----

l := tabName[„Arbeitsschritte“,i][„mögliche Stationen“,tabName[„Arbeitsschritte“,i].ydim].ydim + 1;

XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/ProcessPlan[„,i,“]/Process[„,j,“]/
ResourcesRequired/Resource/ResourceIdentifier));
XMLSchnittstelle.holeNächstenKnoten;
tabName[„Arbeitsschritte“,i][„mögliche Stationen“,tabName[„Arbeitsschritte“,i].ydim][„Ressource“,l] :=
Hilfsmethoden.Liefere_Name („Resource“,XMLSchnittstelle.holeKnotenWert,true);

/* Einlesen der Bearbeitungszeit */
-----

XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/ProcessPlan[„,i,“]/Process[„,j,“]/
OperationTime/Value));
XMLSchnittstelle.holeNächstenKnoten;
tabName[„Arbeitsschritte“,i][„mögliche Stationen“,tabName[„Arbeitsschritte“,i].ydim][„Bearbeitungszeit“,l]
:= str_to_time(XMLSchnittstelle.holeKnotenWert);

XMLSchnittstelle.schliesseKinder;
end;
end;
XMLSchnittstelle.schliesseKinder;
end;
XMLSchnittstelle.schließen;
end;

```

B.2 Methode zum Einlesen der Ressourcen und Materialflussobjekte

```

/* Importmethode des Modellgenerators
Daten werden durch eine CMSD-XML-Datei (Spezifikation Stand 30.04.2010) bereitgestellt.
Einlesen der Ressourcenobjekte „Machine“, der Pufferobjekte sowie Quellen
und Senken in die Verwaltungstabellen Maschinendaten und Materialflussobjekte. */
-----

(Pfad_XML_Datei:string)
is
    tabName
        : object;
    i,j,k,l,m,XPos,YPos
        : integer;
    tmpStr,tempStrAnh
        : string;
    tmpTeileTyp,tmpSetup
        : string;
    tempTime
        : time;
do
    Maschinendaten.loeschen({0,1}..{Maschinendaten.xdim,*});
    Materialflussobjekte.loeschen({0,1}..{Materialflussobjekte.xdim,*});
    XMLSchnittstelle.öffneDokument;
    XMLSchnittstelle.selektiereKnoten („/CMSDDocument/DataSection/Resource");
    i := 0;

    while XMLSchnittstelle.holeNächstenKnoten loop
        XMLSchnittstelle.selektiereKinder;
        i := i+1;

        /* Überprüfung des Ressourcentypes und Einlesen des Maschinennamen und des Schichtkalenders.
        Selektion der Knoten in der XML-Datei per XPath http://www.w3.org/TR/xpath/ */
        -----

        While XMLSchnittstelle.holeNächstenKnoten loop
            If XMLSchnittstelle.holeKnotenNamen = „ResourceType" then
                Inspect XMLSchnittstelle.holeKnotenWert
                    when „machine" then
                        tabName := Maschinendaten;
                        j := tabName.ydim+1;
                        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Resource[„,i,“]/Name));
                        XMLSchnittstelle.holeNächstenKnoten;
                        tabName[„Name“,j] := XMLSchnittstelle.holeKnotenWert;

                        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Resource[„,i,“]/
ShiftAssignment/CalendarIdentifier));

```



```

XMLSchnittstelle.selektiereKnoten(to_str(„/CMSDDocument/DataSection/SetupDefinition[co
ntains(Identifier,“,to_str(„“,tmpStr,““),““)]/Property[contains(„Ref_Ruestmatrix“,Name)]/SetupChangeoverReference/
SetupChangeoverIdentifier“));

XMLSchnittstelle.holeNaechstenKnoten;
tmpStr := XMLSchnittstelle.holeKnotenWert;

XMLSchnittstelle.selektiereKnoten(to_str(„/CMSDDocument/DataSection/SetupChangeoverDefi
nition[contains(Identifier,“,to_str(„“,tmpStr,““),““)]/NewSetup“));
l := 0;

While XMLSchnittstelle.holeNaechstenKnoten loop
XMLSchnittstelle.selektiereKinder;
l := l+1;

While XMLSchnittstelle.holeNaechstenKnoten loop
XMLSchnittstelle.selektiereKnoten(to_str(„/CMSDDocument/DataSection/SetupChang
eoverDefinition[contains(Identifier,“,to_str(„“,tmpStr,““),““)]/NewSetup[„l,““]/SetupDefinitionIdentifier“));
XMLSchnittstelle.holeNaechstenKnoten;
tmpTeileTyp := XMLSchnittstelle.holeKnotenWert;

XMLSchnittstelle.selektiereKnoten(to_str(„/CMSDDocument/DataSection/SetupChang
eoverDefinition[contains(Identifier,“,to_str(„“,tmpStr,““),““)]/ChangeoverTime/Value“));
XMLSchnittstelle.holeNaechstenKnoten;
tempTime := str_to_time(XMLSchnittstelle.holeKnotenWert);

XMLSchnittstelle.selektiereKnoten(to_str(„/CMSDDocument/DataSection/Set
upDefinition[contains(Identifier,“,to_str(„“,tmpTeileTyp,““),““)]/Property[contains(„Ist_geruestet_fuer“,Name)]/
child::Value“));

XMLSchnittstelle.holeNaechstenKnoten;
tmpTeileTyp := XMLSchnittstelle.holeKnotenWert;
tabName[„Ruestmatrix“,j][TmpTeileTyp,k] := tempTime;

end;

XMLSchnittstelle.schliesseKinder;
end;
end;
XMLSchnittstelle.schliesseKinder;
end;

/* Leere Tabellenfelder mit 0 belegen */
-----
For k := 1 to tabName[„Ruestmatrix“,j].ydim loop
For l := 1 to tabName[„Ruestmatrix“,j].xdim loop
If tabName[„Ruestmatrix“,j][l,k] = void then
tabName[„Ruestmatrix“,j][l,k] := 0;
end;
next;
next;

end;

/* Ressource ist nicht vom Typ „machine“ */
-----
when „other“ then
tabName := Materialflussobjekte;
j := tabName.ydim+1;
XMLSchnittstelle.selektiereKnoten(to_str(„/CMSDDocument/DataSection/Resource[„i,““]/Name“));
XMLSchnittstelle.holeNaechstenKnoten;
tabName[„Name“,j] := XMLSchnittstelle.holeKnotenWert;

/* Einlesen des Materialflussobjekttypes */
-----
XMLSchnittstelle.selektiereKnoten(to_str(„/CMSDDocument/DataSection/Resource[„i,““]/ResourceClass/
ResourceClassIdentifier“));
XMLSchnittstelle.holeNaechstenKnoten;
tabName[„Typ“,j] := XMLSchnittstelle.holeKnotenWert;

/* Einlesen der Kapazität, der Verfügbarkeit und der MTTR, sind in der XML-Datei keine Daten
zur Verfügbarkeit und MTTR vermerkt werden 100 und 0 als Standardwerte gesetzt */
-----
XMLSchnittstelle.selektiereKnoten(to_str(„/CMSDDocument/DataSection/Resource[„i,““]/Property[conta
ins(„Kapazitaet“,Name)]/child::Value“));
XMLSchnittstelle.holeNaechstenKnoten;
tabName[„Kapazität“,j] := str_to_num(XMLSchnittstelle.holeKnotenWert);

XMLSchnittstelle.selektiereKnoten(to_str(„/CMSDDocument/DataSection/Resource[„i,““]/Property[conta
ins(„Verfuegbarkeit“,Name)]/child::Value“));
XMLSchnittstelle.holeNaechstenKnoten;
tabName[„Verfügbarkeit“,j] := str_to_num(XMLSchnittstelle.holeKnotenWert);

If XMLSchnittstelle.holeKnotenWert = „“ then
tabName[„Verfügbarkeit“,j] := 100;
end;

XMLSchnittstelle.selektiereKnoten(to_str(„/CMSDDocument/DataSection/Resource[„i,““]/Property[conta
ins(„MTTR“,Name)]/child::Value“));
XMLSchnittstelle.holeNaechstenKnoten;
tabName[„MTTR“,j] := str_to_time(XMLSchnittstelle.holeKnotenWert);

/* Einlesen der Prioritätsregel des Puffers */
-----
XMLSchnittstelle.selektiereKnoten(to_str(„/CMSDDocument/DataSection/Resource[„i,““]/Property[conta
ins(„Prioritätsregel“,Name)]/child::Value“));
XMLSchnittstelle.holeNaechstenKnoten;
tabName[„Prioritätsregel“,j] := XMLSchnittstelle.holeKnotenWert;

/* Einlesen der Vorgänger und Nachfolger */
-----
XMLSchnittstelle.selektiereKnoten(to_str(„/CMSDDocument/DataSection/Resource[„i,““]/
GroupDefinition/Connection/FromResource/ResourceIdentifier“));
XMLSchnittstelle.holeNaechstenKnoten;

If XMLSchnittstelle.holeKnotenWert /= „“ then
tabName.erzeugeSubListe(6,j);
tabName[„Vorgänger“,j].setzeName(„Vorgängerliste“);
end;

XMLSchnittstelle.selektiereKnoten(to_str(„/CMSDDocument/DataSection/Resource[„i,““]/
GroupDefinition/Connection/ToResource/ResourceIdentifier“));
XMLSchnittstelle.holeNaechstenKnoten;

If XMLSchnittstelle.holeKnotenWert /= „“ then

```



```

        tabName.erzeugeSubListe(7,j);
        tabName[„Nachfolger“,j].setzeName(„Nachfolgerliste“);
    end;

    XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Resource[„,i,“]/
GroupDefinition/Connection“));
    m := 0;

    While XMLSchnittstelle.holeNächstenKnoten loop
        XMLSchnittstelle.selektiereKinder;
        m := m+1;

        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Resource[„,i,“]/
GroupDefinition/Connection[„,m,“]/FromResource/ResourceIdentifier“));
        XMLSchnittstelle.holeNächstenKnoten;

        If XMLSchnittstelle.holeKnotenWert /= „“ then
            tmpStr := Hilfsmethoden.Liefere_Name („Resource“,XMLSchnittstelle.holeKnotenWert,true);
            tabName[„Vorgänger“,j].anhaengen(tmpStr);
        end;

        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Resource[„,i,“]/
GroupDefinition/Connection[„,m,“]/ToResource/ResourceIdentifier“));
        XMLSchnittstelle.holeNächstenKnoten;

        If XMLSchnittstelle.holeKnotenWert /= „“ then
            tmpStr := Hilfsmethoden.Liefere_Name („Resource“,XMLSchnittstelle.holeKnotenWert,true);
            tabName[„Nachfolger“,j].anhaengen(tmpStr);
        end;
        XMLSchnittstelle.schliesseKinder;
    end;

    end;
end;
XMLSchnittstelle.schliesseKinder;
end;

/* Einlesen der Standortkoordinaten */
-----
XMLSchnittstelle.selektiereKnoten („/CMSDDocument/DataSection/Layout“);
tabName := void;
i := 0;
j := 0;

while XMLSchnittstelle.holeNächstenKnoten loop
    XMLSchnittstelle.selektiereKinder;
    i := i+1;

    XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Layout[„,i,“]/Placement“));

    While XMLSchnittstelle.holeNächstenKnoten loop
        XMLSchnittstelle.selektiereKinder;
        j := j+1;
        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Layout[„,i,“]/Placement[„,j,“]/
LayoutElementIdentifier“));
        XMLSchnittstelle.holeNächstenKnoten;
        tmpStr := XMLSchnittstelle.holeKnotenWert;

        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Layout[„,i,“]/Placement[„,j,“]/
Location/X“));
        XMLSchnittstelle.holeNächstenKnoten;
        XPos := str_to_num(XMLSchnittstelle.holeKnotenWert);

        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Layout[„,i,“]/Placement[„,j,“]/
Location/Y“));
        XMLSchnittstelle.holeNächstenKnoten;
        YPos := str_to_num(XMLSchnittstelle.holeKnotenWert);

        /* Überprüfung des Ressourcentyps */
        -----
        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/LayoutObject[contains („,to_str („\",tmpStr
,\"\"),„,Identifier)/Type“));
        XMLSchnittstelle.holeNächstenKnoten;

        Inspect XMLSchnittstelle.holeKnotenWert
            when „machine“ then
                tabName := Maschinendaten;

            when „other“ then
                tabName := Materialflussobjekte;
        end;

        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/LayoutObject[contains („,to_str („\",tmpStr
,\"\"),„,Identifier)/AssociatedResource/ResourceIdentifier“));
        XMLSchnittstelle.holeNächstenKnoten;
        tmpStr := Hilfsmethoden.Liefere_Name („Resource“,XMLSchnittstelle.holeKnotenWert,true);

        If tabName /= void then
            tabName[„X_Position“,tmpStr] := XPos;
            tabName[„Y_Position“,tmpStr] := YPos;
        end;
        XMLSchnittstelle.schliesseKinder;
    end;
    XMLSchnittstelle.schliesseKinder;
end;
XMLSchnittstelle.schließen;

end;

```

B.3 Methode zum Einlesen des Schichtkalenders

```

/* Schnittstellenmethode zwischen der CMSD-XML-Datei (Spezifikation Stand 30.04.2010) und dem
Modellgenerator. Erzeugen und einlesen der Schichtkalender in der CMSD-Schnittstelle. */
-----

(Pfad_XML_Datei:string;Pfad_ArbFlaeche:object)

is
  tempSchichtplan          : table;
  tempPausen               : list[string];
  obj,objTyp               : object;
  i,j,k,xZeitEinheit,strPos : integer;
  tmpReal                  : real;
  tmpTime                  : time;
  tmpStr,tempStd            : string;

do
  /* Löschen von evtl. vorhandenen Schichtkalenderobjekten im Netzwerk Werkstattfertigung */
  -----
  objTyp := ~.Objekte.Schichtkalender;

  For i := Pfad_ArbFlaeche.AnzahlObjekte downto 1 loop
    If Pfad_ArbFlaeche.Objekt(i).Klasse = objTyp then
      Pfad_ArbFlaeche.Objekt(i).vernichteObjekt;
    end;
  next;

  XMLSchnittstelle.öffneDokument;
  XMLSchnittstelle.selektiereKnoten („/CMSDDocument/DataSection/Calendar");
  i := 0;

  While XMLSchnittstelle.holeNächstenKnoten loop
    XMLSchnittstelle.selektiereKinder;
    tempSchichtplan.create;
    tempSchichtplan.setzeDatentyp(1,11,"string");
    tempSchichtplan.setzeDatentyp({2,*},{3,*},"time");
    tempSchichtplan.setzeDatentyp({4,*}..{10,*},"boolean");

    j := 0;
    i := i+1;

    While XMLSchnittstelle.holeNächstenKnoten loop

      /* Erstellen eines Schichtkalenders im Netzwerk „Werkstattfertigung“ für jedes Calendarobjekt in der
      CMSD-XML-Datei. */
      -----
      XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Calendar["&i;"]/Identifier"));
      XMLSchnittstelle.holeNächstenKnoten;

      tmpStr := XMLSchnittstelle.holeKnotenWert;
      objTyp := ~.Objekte.Schichtkalender;
      obj := objTyp.erzeugeObjekt(Pfad_ArbFlaeche,40,100+60*(i-1),tmpStr);

      XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Calendar["&i;"]/Shift"));

      while XMLSchnittstelle.holeNächstenKnoten loop
        XMLSchnittstelle.selektiereKinder;
        j := j+1;

        /* Einlesen des Schichtnamens und der Startzeit der Schicht */
        -----
        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Calendar["&i;"]/Shift["&j;"]/
Identifier"));
        XMLSchnittstelle.holeNächstenKnoten;
        tempSchichtplan[1,j] := XMLSchnittstelle.holeKnotenWert;

        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Calendar["&i;"]/Shift["&j;"]/
StartTime"));
        XMLSchnittstelle.holeNächstenKnoten;
        tempSchichtplan[2,j] := str_to_time(XMLSchnittstelle.holeKnotenWert);

        /* Auslesen der Schichtdauer und Berechnung des Schichtendes */
        -----
        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Calendar["&i;"]/Shift["&j;"]/Duration/
Value"));
        XMLSchnittstelle.holeNächstenKnoten;
        tmpReal := str_to_num(XMLSchnittstelle.holeKnotenWert);

        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Calendar["&i;"]/Shift["&j;"]/Duration/
Unit"));
        XMLSchnittstelle.holeNächstenKnoten;

        Inspect XMLSchnittstelle.holeKnotenWert
          when „hour“ then
            xZeitEinheit := 3600;
          when „minute“ then
            xZeitEinheit := 60;
          when „second“ then
            xZeitEinheit := 1;
        end;

        If (tempSchichtplan[2,j] + tmpReal * xZeitEinheit) >= str_to_time („1:00:00:00“) then
          tempSchichtplan[3,j] := (tempSchichtplan[2,j] + tmpReal * xZeitEinheit) - str_to_time („1:00:00:00");
        Else
          tempSchichtplan[3,j] := tempSchichtplan[2,j] + tmpReal * xZeitEinheit;
        end;

        /* Einlesen der der Schicht zugewiesenen Tage */
        -----
        XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Calendar["&i;"]/Shift["&j;"]/
ApplicableDay"));
        k := 0;

        While XMLSchnittstelle.holeNächstenKnoten loop
          k := k+1;
          XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Calendar["&i;"]/Shift["&j;"]/
ApplicableDay["&k;"]));
          Inspect XMLSchnittstelle.holeKnotenWert
            when „monday“ then
              tempSchichtplan[4,j] := true;

```

```

        when „tuesday“ then
            tempSchichtplan[5,j] := true;
        when „wednesday“ then
            tempSchichtplan[6,j] := true;
        when „thursday“ then
            tempSchichtplan[7,j] := true;
        when „friday“ then
            tempSchichtplan[8,j] := true;
        when „saturday“ then
            tempSchichtplan[9,j] := true;
        when „sunday“ then
            tempSchichtplan[10,j] := true;
    end;
end;

/* Einlesen der Pausenzeiten:
1. Einlesen der Startzeit der Pause
2. Einlesen der Pausendauer im Dezimalzahlenformat!
3. Fallunterscheidung der Zeiteinheit der Pausendauer (Stunden, Minuten, Sekunden)
Ziel: Überführung der CMSD-Pausenzeitbeschreibung in die Tabellenübersicht des
Schichtkalenders in Plant Simulation

Hierfür notwendig: a) Ermittlung des Endzeitpunktes der Pause mittels der Hilfsvariablen „tmpReal“
und der Hilfsvariable „xZeitEinheit“ als Faktor für die Zeitheit
b) Erstellung eines Strings der Pausenzeiten gemäß des Formates im Schichtkalender*/
-----
XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Calendar[„,i,“]/Shift[„,j,“]/Break"));
tempPausen.create;
k := 0;

While XMLSchnittstelle.holeNächstenKnoten loop
    XMLSchnittstelle.selektiereKinder;
    k := k+1;

    XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Calendar[„,i,“]/Shift[„,j,“]/
Break[„,k,“]/StartTime"));
    XMLSchnittstelle.holeNächstenKnoten;
    tmpTime := str_to_time(XMLSchnittstelle.holeKnotenWert);

    XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Calendar[„,i,“]/Shift[„,j,“]/
Break[„,k,“]/Duration/Value"));
    XMLSchnittstelle.holeNächstenKnoten;
    tmpReal := str_to_num(XMLSchnittstelle.holeKnotenWert);

    XMLSchnittstelle.selektiereKnoten(to_str („/CMSDDocument/DataSection/Calendar[„,i,“]/Shift[„,j,“]/
Break[„,k,“]/Duration/Unit"));
    XMLSchnittstelle.holeNächstenKnoten;

    Inspect XMLSchnittstelle.holeKnotenWert
        when „hour“ then
            xZeitEinheit := 3600;
        when „minute“ then
            xZeitEinheit := 60;
        when „second“ then
            xZeitEinheit := 1;
    end;

    /* Bedingung für das Subtrahieren eines Tages bei einer Pause, welche von einem Zeitpunkt
    des Tages bis in den darauffolgenden Tag hinein dauert (Bsp. 23:45-00:15).

    zu b) Die Formatabpassung erfolgt mittels Stringfunktionen die die Zeichenfolge „:00.0000“
    von den Zeitangaben abtrennen. Jede Pausenangabe wird zunächst in der Liste „tempPausen“
    aufgenommen und später zu einem String mit dem Trennzeichen „;“ zusammengefasst. */
    -----

    strPos := pos („.",time_to_str(tmpTime))-3;
    If strPos < 5 then
        tempStd := „0:“
    Else
        tempStd := „“;
    end;

    If (tmpTime + tmpReal * xZeitEinheit) >= str_to_time („1:00:00:00“) then

        tempPausen[k] := tempStd + omit(time_to_str(tmpTime),strPos,8) + „-“;
        tmpStr := time_to_str((tmpTime + tmpReal * xZeitEinheit) - str_to_time („1:00:00:00"));
        strPos := pos („.",tmpStr)-3;

        If strPos < 5 then
            tempStd := „0:“
        Else
            tempStd := „“;
        end;

        tmpStr := tempStd + omit(tmpStr,strPos,8);
        tempPausen[k] := tempPausen[k] + tmpStr;

    Else
        tempPausen[k] := tempStd + omit(time_to_str(tmpTime),strPos,8) + „-“;
        tmpStr := time_to_str(tmpTime + tmpReal * xZeitEinheit);
        strPos := pos („.",tmpStr)-3;

        If strPos < 5 then
            tempStd := „0:“
        Else
            tempStd := „“;
        end;

        tmpStr := tempStd + omit(tmpStr,strPos,8);
        tempPausen[k] := tempPausen[k] + tmpStr;
    end;

    /* Einfügen des Trennzeichens „;“ an die Pause, wenn in dieser Schicht noch mindestens eine
    weitere Pause folgt. */
    -----

    If k > 1 then
        tempPausen[k-1] := tempPausen[k-1] + „;“;
    end;
    XMLSchnittstelle.schliesseKinder;
end;

/* Liste tempPausen zu einem String der Pausenzeiten mit Trennzeichen „;“ vereinigen */
-----

```

```

        For k:=1 to tempPausen.dim loop
            tempSchichtplan[11,j] := tempSchichtplan[11,j] + tempPausen.entnehmen(1);
        next;
        XMLSchnittstelle.schliesseKinder;
    end;
end;
XMLSchnittstelle.schliesseKinder;
obj.Schichtplan := tempSchichtplan;
end;
XMLSchnittstelle.schließen;
end;

```

C Ausgewählte Strukturen der CMSD-Eingangsdaten des Beispielszenarios in XML

C.1 Auszug zur Definition einer Maschine

```

<Resource>
  <Identifier>Werk01_M1</Identifier>
  <Description>Gewindedrehmaschine</Description>
  <ResourceType>machine</ResourceType>
  <ResourceClass>
    <ResourceClassIdentifier>Klasse_Drehmaschinen</ResourceClassIdentifier>
  </ResourceClass>
  <Name>M1</Name>
  <ShiftAssignment>
    <CalendarIdentifier>Schichtkalender</CalendarIdentifier>
  </ShiftAssignment>
  <GroupDefinition>
    <Connection>
      <Identifier>String</Identifier>
      <Description>String</Description>
      <FromResource>
        <ResourceIdentifier>Werk01_Eing_Dreh</ResourceIdentifier>
      </FromResource>
    </Connection>
    <Connection>
      <Identifier>String</Identifier>
      <Description>String</Description>
      <ToResource>
        <ResourceIdentifier>Werk01_Ausg_Dreh</ResourceIdentifier>
      </ToResource>
    </Connection>
  </GroupDefinition>
  <Property>
    <Name>Verfuegbarkeit</Name>
    <Description>Maschinenverfuegbarkeit</Description>
    <Unit>Prozent</Unit>
    <Value>96</Value>
  </Property>
  <Property>
    <Name>MTTR</Name>
    <Description>Mittlere Reparaturzeit</Description>
    <Unit>Minuten</Unit>
    <Value>1:10</Value>
  </Property>
  <Property>
    <Name>Maschinenzuverlaessigkeit</Name>
    <Description>Anteil der fehlerfreien Fertigung in Bezug zur Gesamtfertigung</Description>
    <Unit>Prozent</Unit>
    <Value>99.5</Value>
  </Property>
  <Property>
    <Name>Ruestmatrix</Name>
    <Description>Referenz zur Ruestmatrix von M1</Description>
    <SetupDefinitionReference>
      <SetupDefinitionIdentifier>Setup_M1</SetupDefinitionIdentifier>
    </SetupDefinitionReference>
  </Property>
</Resource>

```

C.2 Auszug zur Definition der Rüstmatrix einer Maschine

```

<Resource>
  <Identifier>Werk01_M1</Identifier>
  <Description>Gewindedrehmaschine</Description>
  <ResourceType>machine</ResourceType>
  <ResourceClass>
    <ResourceClassIdentifier>Klasse_Drehmaschinen</ResourceClassIdentifier>
  </ResourceClass>
  <Name>M1</Name>
  ...
  <Property>
    <Name>Ruestmatrix</Name>
    <Description>Referenz zur Ruestmatrix von M1</Description>
    <SetupDefinitionReference>
      <SetupDefinitionIdentifier>Setup_M1</SetupDefinitionIdentifier>
    </SetupDefinitionReference>
  </Property>
</Resource>

<SetupDefinition>
  <Identifier>Setup_M1</Identifier>
  <Description>Rüstzeit Maschine 1</Description>
  <SetupResource>
    <ResourceIdentifier>M1</ResourceIdentifier>
  </SetupResource>
  <Name>Setup_Maschine1</Name>
  <ChildSetup>
    <SetupDefinitionIdentifier>Setup_M1_Teil1</SetupDefinitionIdentifier>
  </ChildSetup>

```

```

    </ChildSetup>
  </ChildSetup>
  <SetupDefinitionIdentifier>Setup_M1_Teil2</SetupDefinitionIdentifier>
</ChildSetup>
...
</SetupDefinition>

<SetupDefinition>
  <Identifier>Setup_M1_Teil1</Identifier>
  <Description>Rüstzeit Maschine 1, aktuelles Setup Teil 1</Description>
  <SetupResource>
    <ResourceIdentifier>M1</ResourceIdentifier>
  </SetupResource>
  <Name>String</Name>
  <Property>
    <Name>Ist geruestet fuer</Name>
    <Description>Teiletyp für den die Maschine gerüstet ist</Description>
    <Unit>Name</Unit>
    <Value>Teil_1</Value>
  </Property>
  <Property>
    <Name>Ref Ruestmatrix</Name>
    <Description>Referenz zur Rüstmatrix Zeile Teil1</Description>
    <SetupChangeoverReference>
      <SetupChangeoverIdentifier>Konfig_M1_Teil1</SetupChangeoverIdentifier>
    </SetupChangeoverReference>
  </Property>
</SetupDefinition>

<SetupChangeoverDefinition>
  <Identifier>Konfig_M1_Teil1</Identifier>
  <Description>Setupwechsel von Setup Teil 1 to Teil x</Description>
  <CurrentSetup>
    <SetupDefinitionIdentifier>Setup_M1_Teil1</SetupDefinitionIdentifier>
  </CurrentSetup>
  <NewSetup>
    <SetupDefinitionIdentifier>Setup_M1_Teil2</SetupDefinitionIdentifier>
    <ChangeoverTime>
      <TimeUnit>second</TimeUnit>
      <Value>12</Value>
    </ChangeoverTime>
  </NewSetup>
  <NewSetup>
    <SetupDefinitionIdentifier>Setup_M1_Teil3</SetupDefinitionIdentifier>
    <ChangeoverTime>
      <TimeUnit>second</TimeUnit>
      <Value>15</Value>
    </ChangeoverTime>
  </NewSetup>
  <NewSetup>
    <SetupDefinitionIdentifier>Setup_M1_abgeruestet</SetupDefinitionIdentifier>
    <ChangeoverTime>
      <TimeUnit>second</TimeUnit>
      <Value>30</Value>
    </ChangeoverTime>
  </NewSetup>
</SetupChangeoverDefinition>

```